Air Force Institute of Technology
AFIT Scholar

3-11-2011

Defensive Cyber Battle Damage Assessment Through Attack Methodology Modeling

Ryan T. Ostler

Follow this and additional works at: https://scholar.afit.edu/etd

Part of the Computer and Systems Architecture Commons, and the Information Security Commons

DEFENSIVE CYBER BATTLE DAMAGE ASSESSMENT
THROUGH ATTACK METHODOLOGY MODELING

THESIS

Ryan T. Ostler, Captain, USAF

AFIT/GCO/ENG/11-11

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT/GCO/ENG/11-11

DEFENSIVE CYBER BATTLE DAMAGE ASSESSMENT

THROUGH ATTACK METHODOLOGY MODELING

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Engineering

Ryan T. Ostler, B.S. Computer Science
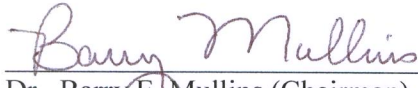Captain, USAF

March 2011

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/GCO/ENG/11-11

DEFENSIVE CYBER BATTLE DAMAGE ASSESSMENT

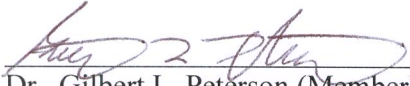THROUGH ATTACK METHODOLOGY MODELING

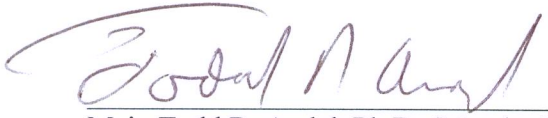Ryan T. Ostler, B.S. Computer Science
Captain, USAF

Approved:

_____
Dr. Barry E. Mullins (Chairman)

21 Mar 11
_____
Date

_____
Dr. Gilbert L. Peterson (Member)

21 MAR 11
_____
Date

_____
Maj. Todd R. Andel, Ph.D. (Member)

21 MAR 11
_____
Date

AFIT/GCO/ENG/11-11

Abstract

Due to the growing sophisticated capabilities of advanced persistent cyber threats, it is necessary to understand and accurately assess cyber attack damage to digital assets. This thesis proposes a Defensive Cyber Battle Damage Assessment (DCBDA) process which utilizes the comprehensive understanding of all possible cyber attack methodologies captured in a Cyber Attack Methodology Exhaustive List (CAMEL). This research proposes CAMEL to provide detailed knowledge of cyber attack actions, methods, capabilities, forensic evidence and evidence collection methods. This product is modeled as an attack tree called the Cyber Attack Methodology Attack Tree (CAMAT). The proposed DCBDA process uses CAMAT to analyze potential attack scenarios used by an attacker. These scenarios are utilized to identify the associated digital forensic methods in CAMEL to correctly collect and analyze the damage from a cyber attack. The results from the experimentation of the proposed DCBDA process show the process can be successfully applied to cyber attack scenarios to correctly assess the extent, method and damage caused by a cyber attack.

Acknowledgments

I would like to thank my advisor Dr. Barry Mullins for his immeasurable patience with me during my deluge of thesis woes. I would also like to thank my committee members, Dr. Gilbert Peterson and Maj Andel, for their guidance during my research. A special thanks to my fellow cyber Airman Captain Owen for all the support through this process. Finally, no words can express my appreciation to my wife and children.

Table of Contents

List of Figures

List of Tables

DEFENSIVE CYBER BATTLE DAMAGE ASSESSMENT

THROUGH ATTACK METHODOLOGY MODELING

I. Introduction

"If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not your enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle."

- Sun Tzu

1.1. Motivation

In all manners of warfare, one must know how the enemy can attack in order to defend. This knowledge can be gained through warfare planning which considers all aspects of the enemy's capabilities and methods of attack. This planning is accomplished through the detailed analysis of the tactics, techniques and procedures (TTPs) an attacker has at their disposal. Comprehension of an adversary's attack methodology is pivotal for successful warfare. Without knowing how an enemy can attack, one cannot effectively plan to posses the ability to defend against the attack.

Motivated attackers exist and have the means to penetrate and cause damage to a computer network. A recent example of this nefarious ability is the 2009 cyber attack on

1

Google. This highly sophisticated and narrowly targeted attack on Google's corporate infrastructure originated from China and resulted in the theft of intellectual property [Dru10]. This attack, codenamed Operation Aurora, utilized a previously unknown vulnerability exploit in Microsoft Internet Explorer aimed at specific targets through tailored emails [Mca10]. The exploit then downloaded and activated malware within the systems. Malware is defined as software designed and/or deployed by adversaries without the consent or knowledge of the user in support of adversarial missions [DOD09]. The attack, which was initiated surreptitiously when targeted users accessed a reputable appearing web page, ultimately connected those computer systems to a remote malicious server. The targeted nature of the attack actions and level of attack method sophistication of Operation Aurora are what garnered the attacker's success. This attack also raised awareness to the advanced abilities of today's cyber attacker.

Another example of a sophisticated targeted cyber attack is Stuxnet. Stuxnet was a malware attack which targeted a specific industrial control system, likely in Iran [FMC10] [MRH10]. These control systems operate critical national assets such as gas pipelines or power plants. The apparent goal of Stuxnet was to sabotage the industrial facility by modifying the control systems to operate outside of their operational limits resulting in damage to the facility's capabilities [FMC10]. Stuxnet is considered the most technologically sophisticated malware developed to date [MRH10]. This malware shows how modern cyber attackers are using targeted, sophisticated and capable attack methodologies.

These examples show that anyone who uses computer networks must assume that exploitable vulnerabilities exist and will be leveraged for unauthorized use. Simply put,

2

the enemy *will* attack, and *will* have success. Continued network operational success is only possible through the ability to identify an attack, assess damage and defend against attacks.

Cyber attack damage assessment forms the foundation for the key mission impact questions of today's information technology-dependant leaders. Leaders need to know the impact of an attack on the assets they control. This impact assessment depends on accurate and timely information from a detailed technical damage assessment of an attack. The attack assessment must pinpoint exactly what happened to the friendly targeted systems to ensure the mission impact derived from the assessment is as accurate as possible.

Knowing how the enemy can attack before they do so allows for a targeted detailed assessment of damage after the attack has occurred. The intelligence preparation of damage assessment through cyber attack methodology analysis is the most reliable way to conduct active cyber damage assessments. These defensive assessments identify technical damage post cyber attack. This attack assessment process, known as Defensive Cyber Battle Damage Assessment (DCBDA), is vital.

1.2. Research Statement

The goal of this research is to provide robust DCBDA capabilities through digital forensic analysis tailored by an understanding of known cyber attacks. The ability to detect and assess an attack at any level of progress is vital to the successful operation of a digital network. The studied abilities of the sophisticated cyber attacker show the likely probability that attacks will have some level of success. It is crucial for an organization

3

to possess the ability to determine what damage a cyber attack has caused to assets in their control. This critical need to perform accurate, timely and comprehensive cyber attack damage assessment is the motivation of this research.

1.3. Research Approach

The ability to provide robust DCBDA can best be accomplished through a thorough, complete understanding of the attacker methodology. This research proposes a DCBDA process which is comprised of two phases: a preparation phase and a forensic analysis phase.

The first phase relies on the defender's preparatory ability to understand, model, translate and analyze the actions and methods of the cyber attacker. The purpose of this phase is to use the prepared comprehensive knowledge of the complete cyber attack methodology to pinpoint exact information needed to correctly identify damage for an active attack event assessment.

The second phase of the DCBDA process is forensic analysis. This phase uses the information identified in the preparation phase to conduct a digital forensic analysis to identify evidence of actual attack damage. Cross-referencing identified evidence with known cyber attack information will reveal probable attack techniques. Once the most probable attack techniques are identified, DCBDA can be much more accurately estimated and reported.

In order for the preparation phase of the DCBDA process to contain the comprehensive knowledge of the actual attack under assessment, this research proposes the Cyber Attack Methodology Exhaustive List (CAMEL). CAMEL is the detailed

4

comprehensive listing of the complete cyber attack methodology to include every known attack action, attack action capability metric, attack method, forensic evidence marker, and forensic evidence collection method. This data is collected from researching the known set of all cyber attacker TTPs. Possible sources for the attacker TTPs are forensic analysis white papers, ethical hacker documentation, real world reports, and personal experiences. Data in the CAMEL is used to create an attack tree modeled as the Cyber Attack Methodology Attack Tree (CAMAT). The CAMAT is used by the DCBDA to analyze attack scenarios and possible evidence for an attack event. The diagram in Figure 1 is a high level view of the phases of the DCBDA process which shows the data flow from the CAMEL/CAMAT.
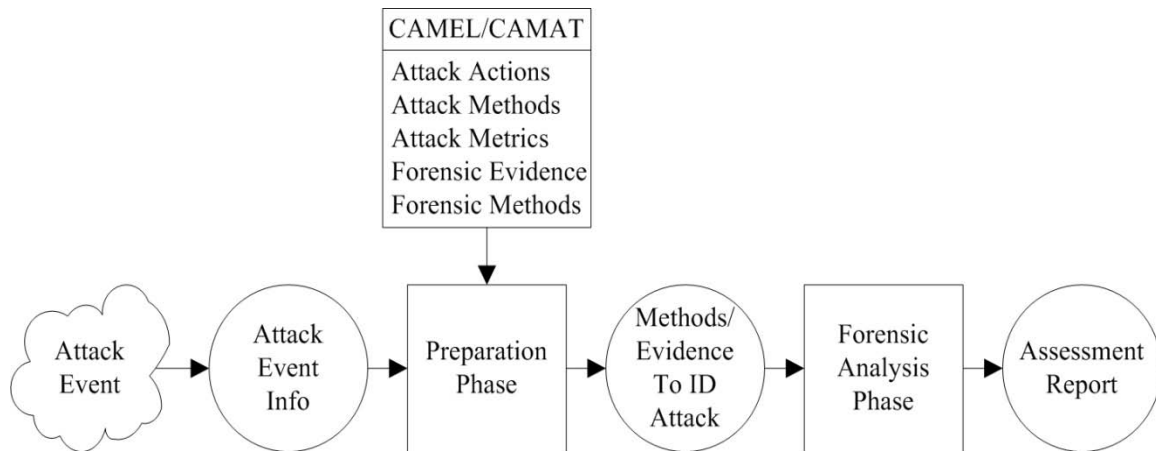
Figure 1. Proposed DCBDA Process.

This approach is broken down into three key objectives for focused effort:

(1) Develop CAMEL and CAMAT creation process

(2) Develop the DCBDA process

(3) Verify the DCBDA process

5

1.3.1. *Develop CAMEL and CAMAT Creation Process*

This objective develops the process to create the CAMEL and CAMAT products. This process outlines the steps required to collect, record and analyze the data which make up these products.

1.3.2. *Develop the DCBDA Process*

This objective seeks to design a defensive cyber technical battle damage assessment process which utilizes the CAMEL and CAMAT products. This process outlines the steps required to assess an attack event which identifies the forensic evidence of the attack's technical battle damage.

1.3.3. *Verify the DCBDA Process*

This objective seeks to verify the success the DCBDA process proposed in this thesis by performing DCBDA against experimental test systems. Experimental success is represented by the degree to which evidence gathered from the assessment correctly identifies the cyber attacks executed against the test systems. Success demonstrates that the proposed DCBDA process can reliably answer the following questions:

- Did an attack occur?
- What potential impact did the attack have?
- What TTPs were used in the attack?

6

1.4. Thesis Organization

Chapter 1 presents an introduction to the research effort, including topic motivation, research statement, research approach and objectives, and the document's organization.

Chapter 2 provides background information on computer security incident analysis and the forensic analysis processes and types. Also discussed are computer taxonomies, the Intelligence Preparation of the Operational Environment (IPOE) process, attack modeling and related work.

Chapter 3 describes the experimental methodology. First, the methodology overview is covered. Next, the approach to build the CAMEL and CAMAT is described. Then the approach for the Defensive Cyber Battle Damage Assessment process is detailed. The design for the experiment methodology to test DCBDA is then discussed.

Chapter 4 presents the details of the experiments and results. This Section covers the creation of CAMEL and CAMAT, the DCBDA process and the verification of the DCBDA process through experimentation.

Chapter 5 contains a summary of the results, conclusions made from the study as well as recommendations for future work.

II. Background

This chapter covers the fundamental concepts and related work of cyber battle damage assessment research. Section 2.1 details the steps involved in a computer security incident analysis. Section 2.2 discusses the forensic analysis processes and types. Section 2.3 discusses different computer attack taxonomies. Section 2.4 covers the Intelligence Preparation of the Operational Environment (IPOE) process. Section 2.5 discusses attack modeling. Section 2.6 discusses related work.

2.1. Computer Security Incident Analysis Process

There are many definitions on what comprises a computer security incident and the process to handle the event. For the purposes of this thesis, the reference source for this topic will be the Chairman of the Joint Chiefs of Staff Manual 6510.01A, Information Assurance and Computer Network Defense (CND) Volume I (Incident Handling Program) which provides guidance for the Department of Defense (DOD) Incident Handling Program [DOD09]. This document acknowledges the adversary threat cannot be completely eliminated and will likely evolve over time. Therefore it is crucial to maintain a proactive, progressive, and coordinated approach to detecting and responding to events and incidents that can adversely affect DOD networks and systems [DOD09].

The DOD Incident Handling Program is part of the overall CND actions to protect, monitor, analyze, detect and respond to unauthorized activity within the DOD computer networks. The program outlines the methodology to provide a general

8

standardized process that establishes the intent and requirements for detecting, analyzing, and responding to information or technology events or incidents for the purpose of mitigating any adverse operational or technical impact on DOD critical data assets, systems, and networks. [DOD09]. Part of the incident handling methodology process is the incident analysis phase which is discussed in this Section.

2.1.1. *Incident Analysis*

Incident analysis is a series of analytical steps taken to identify what happened from an incident. The purpose of incident analysis is to determine and characterize the technical details, root causes, and potential impact of the incident. This understanding will help determine what additional information to gather, coordinate information sharing with others, and develop a course of action for response. This activity relies on effective acquisition, preservation, and timely reporting of incident data [DOD09]. The primary objectives for this phase include:

- Identifying the root cause(s) of the incident through technical analysis

- Ensuring the accuracy and completeness of incident reports

- Characterizing and communicating the potential impact of the incident

- Systematically capturing the methods used in the attack and the security controls that could prevent future occurrences

- Researching actions that can be taken to respond to and eradicate the risk and/or threat

- Understanding patterns of activity to characterize the threat and direct protective and defensive strategies

9

2.1.2. *Incident Analysis Methodology*

This Section details the steps required for an incident analysis. The steps are: gathering information, validate the incident, determine attack vector, determine system weaknesses, identify root causes, determine impact, research and develop courses of action (COAs), coordinate with others and perform correlation and trending.

2.1.2.1. *Gather information*

In this step the objective is to identify and collect all relevant information about the incident for use in the analysis of the incident [DOD09]. Information gathered may include data previously acquired and preserved, external logs, personal accounts, all-source intelligence, technical information, or the current operational situation.

2.1.2.2. *Validate the incident*

In this step the objective is to review, corroborate, and apply applicable updates to the reported incident to ensure all information is accurate. Related documents should be reviewed and updated to maintain situational awareness, to add relevant data to incomplete information, or to fix erroneous information. Report validation may require the review of available sources of information, such as trusted network and system logs of affected systems, to determine if the suspected activities happened as reported. In this step it is also important to verify the incident is categorized properly [DOD09].

2.1.2.3. Determine attack vector(s)

The data gathered in the previous steps is analyzed to determine the attack vector(s) used by the attacker(s). An attack vector is the primary path or method used by the adversary to cause the incident or event to occur. Attack vectors are used to systematically record major classes of attack methodologies used by adversaries, but do not identify the system-specific root causes of an incident [DOD09]. If more than one attack vector is identified, the attack vectors used by the threat actor must be distinguished as either primary or secondary. For example, use of socially engineered e-mail delivering a malicious payload exploiting a known vulnerability that was preventable [DOD09]. The primary attack vector was the payload delivered by the socially engineered secondary attack vector   Attack vectors should be assessed in accordance with Appendix A. This annex describes the some of the possible major categories and sub-categories of attack vectors. The annex should be used for assigning attack vectors to reportable events or incidents [DOD09].

2.1.2.4. Determine system weaknesses

In this step the objective is to analyze the gathered information from the previous steps to determine any underlying system weaknesses, vulnerabilities, or security controls that could have prevented or mitigated the impact of the incident. Identification of system weaknesses is a process used to systematically record and categorize major classes of security controls that could prevent similar incidents from occurring in the future [DOD09].

11

2.1.2.5. Identify root cause(s)

In this step the objective is to analyze the collected information to determine the system-specific base causes of the incident. This identification expands upon the identified attack vectors and system weaknesses by precisely identifying the sets of conditions allowing the incident to occur. For example, an attack vector may identify a system lacking current Operating System (OS) patches. This is useful for correlation and trending, but is insufficient in identifying the specific cause of the incident and preventing against future occurrences. Root cause identification would determine missing patches or system configurations that allowed the incident to occur [DOD09].

2.1.2.6. Determine impact

The objective of this step is to analyze the information gathered to validate and expand the impact assessment. Impact is assessed based on the degree to which an incident or event adversely affects, or has the potential to affect, the successful accomplishment of operational missions of systems and networks. In determining the actual impact, the current and potential impact of the incident or event on the confidentiality, availability, and integrity of organizational operations, organizational assets, or individuals should be considered [DOD09]. Types of impact are either technical or operational.

Technical Impact (TI) refers to the incident's detrimental impact on the technical capabilities of the organization. TI typically refers to impacts on the network or system machines directly or indirectly affected by the incident. Operational Impact (OI) refers to

detrimental impacts on an organization's ability to perform its mission. This may include direct and/or indirect effects that diminish or incapacitate system or network capabilities, the compromise and/or loss of mission critical data, or the temporary or permanent loss of mission critical applications or systems [DOD09].

2.1.2.7. Research and Develop COAs

This step is focused on the identification of actions necessary to respond to the reportable event or incident, fix the system, and assess the risk for the system or network [DOD09]. Here multiple incident response COAs are developed and the best choice is selected.

2.1.2.8. Coordinate with Others

An organization must work with other involved parties to collect additional information, obtain assistance and additional expertise or guidance, and notify appropriate operational and technical channels regarding changes in the status of reportable events, incidents, and incident handling activities. Timely interagency coordination and deconfliction of operations is crucial to conducting an effective incident response [DOD09].

2.1.2.9. Perform Correlation and Trending

In this step the objective is to analyze and identify relationships and trends between incidents in the short term. A special focus on patterns is placed across long term incidents. Effective and complete reporting throughout the incident handling

lifecycle assures that the DOD has the ability to conduct and identify these trends and patterns [DOD09].

2.2.   Digital Forensics Analysis

Digital forensics is the application of science to the identification, collection, examination, and analysis, of data while preserving the integrity of the information and maintaining a strict chain of custody for the data [NIS06]. Chain of custody is a log of evidence possession and correlating time information. The most common goal of performing forensics analysis is to gain a better understanding of an event of interest by finding and analyzing the facts related to that event [NIS06]. Listed in this Section are some of the various types of forensic analysis types and the basic phases of the forensic analysis process.

2.2.1.   *Computer System Forensic Analysis*

System analysis is the identification, acquisition, examination and analysis of all information from or about the affected computer systems to further incident analysis and understand the full scope of the incident [DOD09]. The system information to be analyzed typically includes anything which can help characterize the incident and develop courses of action. Examples of system information are various logs, files, configuration settings, records of currently logged on users, past connections (logins), running processes, open files, and changes to files or system settings (access control lists (ACLs), registries, permissions) [DOD09]. Particular attention must be given to preserving the integrity of the information and maintaining a strict chain of custody.

14

2.2.2. *Malware Forensic Analysis*

Malware analysis is the process of identifying, analyzing, and characterizing reported software suspected of being malicious [DOD09]. This analysis of suspected malicious software is an essential step in determining the full scope of an incident. Malware analysis can be performed at varying degrees of depth in detail. Depending on the complexity of the malware and depth of analysis required, the time necessary to complete the request can vary from minutes to hours to months [DOD09].

2.2.3. *Network Forensic Analysis*

Network forensic analysis is the process of collecting, examining, and interpreting network traffic to identify and respond to events that violate an organization's security policy [DOD09]. This analysis is conducted on the assets and resources attached to the network or the network infrastructure. Network analysis reveals an adversary's use of network resources, uncovers the network interactions that occurred during an incident, and aids in discovering other affected or vulnerable systems.

2.2.4. *Forensic Analysis Process*

This Section describes the basic phases of the forensic analysis process: collection, examination, analysis, and reporting. This process is depicted in Figure 2.

2.2.4.1. *Collection*

The first phase in the forensic analysis process is to identify, label, record, and acquire data from the possible sources of relevant data, while following guidelines and

15

procedures that preserve the integrity of the data [NIS06]. Collection is typically performed in a timely manner because of the likelihood of losing dynamic data such as current network connections, as well as losing data from volatile devices. Data acquisition of the identified data sources for the collection phase of the forensic analysis process should be performed using a three-step process: developing a plan to acquire the data, acquiring the data, and verifying the integrity of the acquired data [NIS06].
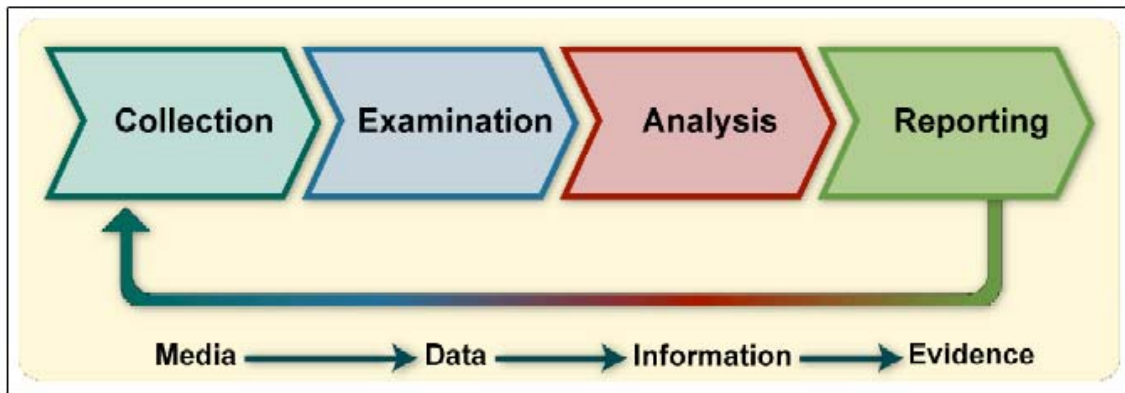
Figure 2. The Forensic Analysis Process [NIS06].

2.2.4.2. Examination

After the data has been collected, the next phase is to examine the data. This involves forensically processing the potentially large amounts of collected data using a combination of automated and manual methods to assess and extract data of particular interest, while preserving the integrity of the data analyzed [NIS06]. The extracted data is then further analyzed using forensic tools and techniques appropriate to the type of information collected.

16

2.2.4.3. Analysis

The next phase of the process is to analyze and interpret the results of the examination, using legally justifiable methods and techniques. The objective of this analysis is to derive useful information which forms conclusions to the questions that were the impetus for performing the collection and examination [NIS06].

2.2.4.4. Reporting

The final phase of the forensic process is reporting the results of the conducted analysis. This report may include describing the actions used and explaining how tools and procedures were selected. Determining what other actions need to be performed such as a forensic examination of additional data sources, securing identified vulnerabilities, and improving existing security controls should also be reported in this phase. The formality and verbosity of the reporting step varies greatly depending on the situation [NIS06].

2.3.   Computer Attack Taxonomy

In the effort to understand the digital computer attack, it is useful to understand the process of the attack which is being studied. In order to do this efficiently, it is worthwhile to classify the distinct elements which make up the attack. This classification allows research efforts to focus on distinct areas of the attack in great detail. A common method for doing this in computer science is with a taxonomy. Taxonomy is the process, or science, of a classification scheme that partitions a body of knowledge and defines the

17

relationship of the pieces [Bos02]. Classification is the process of using a criterion for ordering information into groups or categories. Originally used for classifying organisms, the principles of the classification process apply to computer security incident research via taxonomy. There are many characteristics of taxonomy, and a small set of those is listed below [Amo94]:

- Mutually exclusive. Classification in one category excludes all others

- Exhaustive. Taken together, the categories include all possibilities

- Unambiguous. Data should be clear and precise so that classification is not uncertain, regardless of who is classifying

- Repeatable. Repeated applications result in the same classification

- Accepted. Logical and intuitive so that they could become generally approved through review

- Useful. The process can be used to gain insight into the field of inquiry

2.3.1. *Target-Centric Ontology for Intrusion Detection*

In 2002 Undercoffer and Pinkston from the University of Maryland Baltimore County defined a target-based ontology for intrusion detection [UnP02]. In this paper they devise a simple taxonomy and from it develop their ontology. They argue that ontologies provide software systems with the ability to share a common understanding of the information at issue in turn enabling the software system with a greater ability to reason over and analyze this information [UnP02]. What is unique, outside of their focus on ontology, is that their model is target centric. Their developed ontology is a model of computer attacks categorized by: the system component targeted, the means and

18

consequence of attack, and the location of the attacker [UnP02]. This Section focuses on their efforts to define attack taxonomy.

Their research states that an Intrusion Detection System (IDS) has no knowledge of the attacker's motivation or the tools employed to conduct the attack. The IDS needs to focus on evaluating the attack information which it has the ability to observe. Their taxonomy is classified from the point of view of an IDS according to features and characteristics directly observable at the target. The first section for the taxonomy is the system component targeted by the attack. Table 1 breaks down how the taxonomy defines the categories of targets.

Table 1. Target of Attack Categories [UnP02].

Target	Explanation	Example
Network	The attack is inclusive of the layers of the protocol stack, but does not leave the protocol stack	SynFlood attack
Kernel-Space	A process executing as part of the operating system, either compiled into the kernel or a module that is loaded into and executed by the kernel	Heap overflow
Application	An application running outside of kernel space	Apache Web Server Chunk Handling vulnerability
Other	Any component not included above	Printers, modems, etc.

The next section of the target-centric taxonomy is the means of attack. The means of attack categories are summarized in Table 2. This category is further divided as input validation, exploits and configuration. Input validation vulnerabilities exists if some type of malformed input is received by a hardware or software component and is not properly bounded or checked. Exploits are vulnerabilities such as race conditions or

19

undefined states in a hardware or software component that lead to performance degradation and/or system compromise. Configuration vulnerabilities exist due to an improper state adjustment to a software program.

Table 2. Means of Attack Categories [UnP02].

Category	Type	Description
Input Validation	Buffer Overflow	Overflow of a static-sized data structure.
	Boundary Condition Error	A process attempts to read or write beyond a valid address boundary or a system resource is exhausted
	Malformed Input	A process accepts syntactically incorrect input, extraneous input fields, or the process lacks the ability to handle field-value correlation errors
Exploits	Exceptional Condition	An error resulting from the failure to handle an exceptional condition generated by a functional module or device
	Race Condition	An error occurring during a timing window between two operations
	Serialization Error	An error that results from the improper serialization of operations
	Atomicity Error	An error occurring when a partially-modified data structure is used by another process
Configuration	General	Vulnerabilities that result from some mis-configuration or lack of proper configuration

The taxonomy outlines the consequences and locations of attacks. The categories for the consequences of attacks are listed in Table 3. The consequences of an attack are the end results of the attack.

The location of an attack is the physical location of the attacker. The location categories are outlined in Table 4. These categories are indicated by whether the attacker is connected via the network or local host.

20

Table 3. Consequence Categories [UnP02].

Category	Description
Denial of Service (DoS)	The attack results in a Denial of a Service to the users of the system
User Access	The attack results in the attacker having access to some services on the target system
Root Access	The attack results in the attacker having complete control of the system
Loss of Confidentiality	The attack results in the users of the system loosing privacy of their data
Other	The attack results in the compromise of data integrity or other undesirable characteristic

Table 4. Location Categories [UnP02].

Category	Description
Remote	The attacker does not need to be "virtually" present at the target
Local	The attacker needs to be "virtually" present at the target
Remote/Local	The attacker may be either local or remote to the target

2.3.2. *The Howard Computer and Network Incident Taxonomy*

Starting as a thesis for a PhD at Carnegie Mellon University and later refined as a Sandia National Laboratories report, Howard introduced a computer security incident taxonomy [How97] [HoL98]. Figure 3 represents the entirety of the taxonomy. The taxonomy breaks down a computer and network incident into three main parts; the attacker, the attack(s) and the objective. The taxonomy then further adds fidelity to the nature and makeup of an attack. This taxonomy can be classified as a process schema

21

attack taxonomy. It is important to note that the terms enumerated for the categories in the taxonomy are not intended as a comprehensive dictionary of terms.
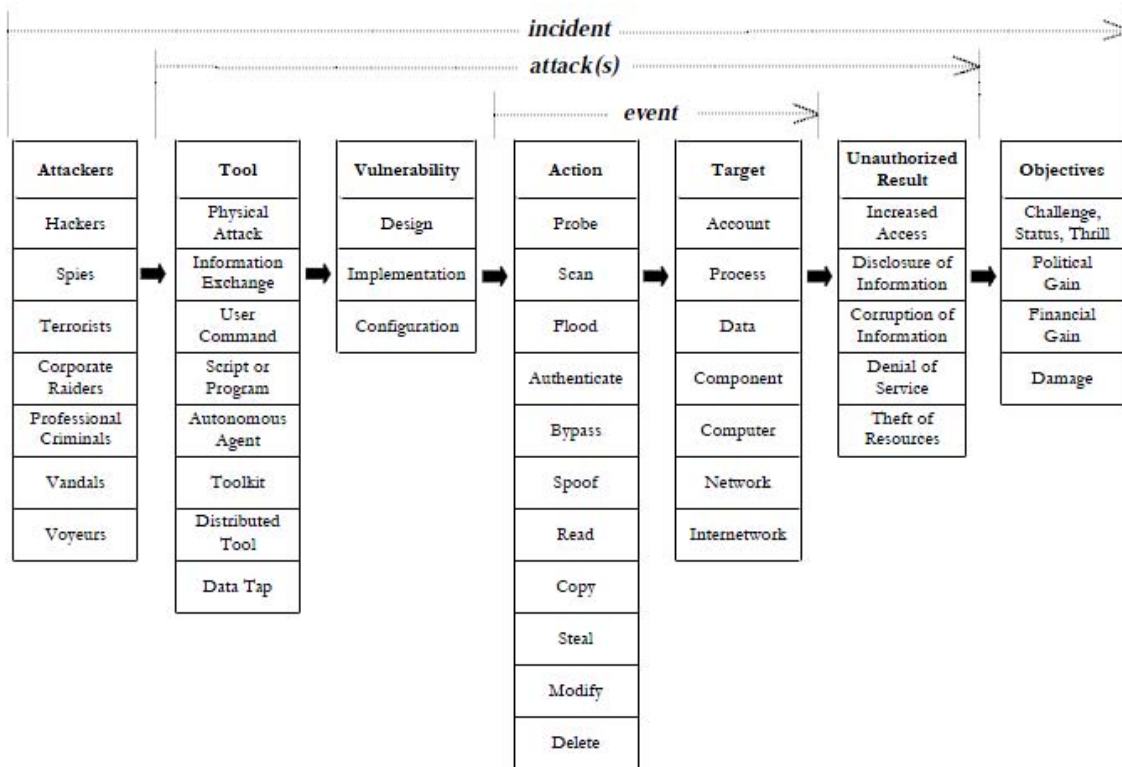
Figure 3. Howard Computer and Network Incident Taxonomy [HoL98].

An attack on a computer or network can be classified as an incident. Figure 4 depicts the three parts of an incident and how one or more attacker achieves their objectives. An incident can be defined as a group of attacks that can be distinguished from other attacks because of the distinctiveness of the attackers, attacks, objectives, sites and timing [HoL98]. The attack may be prosecuted by only one attacker or a group of several attackers related by some objective or other means. From an operational viewpoint, an attacker on a computer attempts to reach their ultimate objectives though

22

some "means, ways, and ends" that connect the attacker to the objective [How97]. The

means used by the attacker are the tools, access and results an attacker achieves.
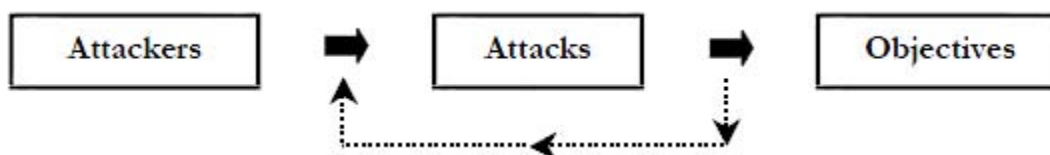
Figure 4. Simplified Computer and Network Incident [HoL98].

2.3.2.1. Attacker

Howard defines an attacker as an individual who attempts one or more attacks in

order to achieve an objective. Table 5 lists the categories of attackers used by Howard.

Table 5. Howard's Attacker Categories [HoL98].

Category	Description
Hackers	For challenge, status or the thrill of obtaining access
Spies	For information to be used for political gain
Terrorists	Cause fear for political gain
Corporate Raiders	Attack competitor's computers for financial gain
Professional Criminals	Personal financial gain
Vandals	Cause damage
Voyeur	Thrill of obtaining sensitive information

23

2.3.2.2. Attack

An attack is a series of steps taken by an attacker to produce an event which achieves an unauthorized result [HoL98]. Howard's taxonomy breaks down an attack into five parts: tool, vulnerability, action, target, and unauthorized result.

The first two steps in an attack, the tool and vulnerability, are used to create an event on a computer or network system. A category listing of tools can is shown in Table 6. A list of vulnerabilities and their descriptions are found in Table 7. During an attack, an attacker uses a tool to exploit a vulnerability that causes a desired action against a target. A tool is some means that can be used to exploit a vulnerability in a computer or network [HoL98]. A tool can be simple or sophisticated in its execution. A vulnerability is some weakness in a system that allows an unintended event or unauthorized action to occur. An attack is comprised of one or more vulnerabilities that are exploited by an attacker using one or more tools.

An event is an action directed at a target which is intended to result in a change of state, or status, of the target [HoL98]. Howard's action categories are found in Table 8. In order for there to be an event, there must be an action that is taken against a target. This brings the breakdown of an event into the separate items of an action and target. An action is a step taken by a user or process in order to achieve a result. The action must be directed against a target, but the action does not have to succeed in changing the state of the target.

24

Table 6. Howard's Tool Categories [HoL98].

Category	Description
Physical attack	A means of physically stealing or damaging a computer, network, its components, or its supporting systems
Information exchange	A means of obtaining information either from other attackers or from the people being attacked
User command	A means of exploiting a vulnerability by entering commands to a process through direct user input at the process interface
Script or program	A means of exploiting a vulnerability by entering commands to a process through the execution of a file of commands (script) or a program at the process interface
Autonomous agent	A means of exploiting a vulnerability by using a program, or program fragment, which operates independently from the user
Toolkit	A software package which contains scripts, programs, or autonomous agents that exploit vulnerabilities
Distributed tool	A tool that can be distributed to multiple hosts, which can then be coordinated to anonymously perform an attack on the target host simultaneously after some time delay
Data tap	A means of monitoring the electromagnetic radiation emanating from a computer or network using an external device

Table 7. Howard's Vulnerability Categories [HoL98].

Category	Description
Design vulnerability	A vulnerability in the design of hardware or software whereby even a perfect implementation will result in a vulnerability
Implementation vulnerability	A vulnerability resulting from an error made in the software or hardware implementation of a satisfactory design
Configuration vulnerability	A vulnerability resulting from an error in the configuration of a system

25

Table 8. Howard's Action Categories [HoL98].

Category	Description
Probe	Access a target in order to determine its characteristics
Scan	Access a set of targets sequentially in order to identify which targets have a specific characteristic
Flood	Access a target repeatedly in order to overload the target's capacity
Authenticate	Present an identity of someone to a process and, if required, verify that identity, in order to access a target
Bypass	Avoid a process by using an alternative method to access a target
Spoof	Masquerade by assuming the appearance of a different entity in network communications
Read	Obtain the content of data in a storage device, or other data medium
Copy	Reproduce a target leaving the original target unchanged
Steal	Take possession of a target without leaving a copy in the original location
Modify	Change the content or characteristics of a target
Delete	Remove a target, or render it irretrievable

A target is some form of data found on a computer or a network system. An event is the logical linkage between an action and a specific target of which an action is directed [HoL98]. A list of target categories is found in Table 9.

The last part of an attack is the result, or more specifically the desired unauthorized result. The possible unauthorized results for Howard's taxonomy are listed in Table 10. An unauthorized result is the logical ending of a successful attack. This is an important distinction in this taxonomy. If the result was authorized it could not be the result of an attack.

Table 9. Howard's Target Categories [HoL98].

Category	Description
Account	A domain of user access on a computer or network which is controlled according to a record of information which contains the user's account name, password and use restrictions
Process	A program in execution, consisting of the executable program, the program's data and stack, its program counter, stack pointer and other registers, and all other information needed to execute the program
Data	Representations of facts, concepts, or instructions in a manner suitable for communication, interpretation, or processing by humans or by automatic means
Component	One of the parts that make up a computer or network
Computer	A device that consists of one or more associated components, including processing units and peripheral units, that is controlled by internally stored programs, and that can perform substantial computations, including numerous arithmetic operations, or logic operations, without human intervention during execution
Network	An interconnected or interrelated group of host computers, switching elements, and interconnecting branches
Internetwork	A network of networks

2.3.2.1. Objective

The objective is the purpose or end goal of an incident. Table 11 depicts Howard's attack taxonomy objective categories. The objective categorizes the efforts, or results, of the attacker's actions.

27

Table 10. Howard's Unauthorized Result Categories [HoL98].

Category	Description
Increased access	An unauthorized increase in the domain of access on a computer or network
Disclosure of information	Dissemination of information to anyone who is not authorized to access that information
Corruption of information	Unauthorized alteration of data on a computer or network
Denial of service	Intentional degradation or blocking of computer or network resources
Theft of resources	Unauthorized use of computer or network resources

Table 11. Howard's Objective Categories [HoL98].

Category	Description
Challenge, status, thrill	No motivation outside of completing attack
Political gain	Attack against a government target
Financial gain	Market advantage from actions on target
Damage	Unauthorized modification or destruction to target

2.3.3. *Landwehr et al: A Taxonomy of Computer Program Security Flaws*

In a paper which conducted a comprehensive survey of computer program security flaw taxonomies, Landwehr et al. proposed a new security flaw taxonomy [LBM93]. The paper describes a taxonomy as not simply a neutral structure for categorizing specimens, but an organized theory of flaws to which we seek answers. From this definition the paper broke down the study of computer attacks into three

28

categories: flaws by genesis, time of introduction, and location [LBM93]. The taxonomy asks these essential questions of each observed flaw, or attack:

- How did it enter the system?

- When did it enter the system?

- Where in the system is it manifest?

2.3.3.1. Flaws By Genesis

The strategies used to avoid, detect or compensate for accidental flaws are different than those introduced intentionally. Table 12 shows the how flaws by genesis may be introduced intentionally or inadvertently [LBM93]. For example, increasing the resources devoted to code reviews and testing may be effective in reducing the number of accidental flaws. Intentional flaws can be categorized as malicious or non-malicious. A malicious flaw is the purposeful advantage of a security flaw in a system. A non-malicious flaw is the intentional introduction of a flaw. This is usually a system service or other computer system function used by a program for non degradation use. An example is a covert channel, which does not harm the host computer system. An inadvertent flaw in software is a flaw that remained undetected through the testing and development phase. There should be no knowledge of this flaw until it is inadvertently discovered.

Table 12. Flaws by Genesis [LBM93].

Genesis	Intentional	Malicious	Trojan Horse
			Trapdoor
			Logic/Time Bomb
		Non-Malicious	Covert Channel
			Other
	Inadvertent	Validation Error	
		Domain Error	
		Serialization/Aliasing	
		Identification/Authentication Inadequate	
		Boundary Condition Violation	
		Other Exploitable Logic Error	

2.3.3.2. Flaws by Time of Introduction

In this Section the software development cycle is explained as a vector for flaws as shown in Table 13. This cycle gives a time factor to how software is produced and an understanding on when certain flaws may occur. The paper introduces three distinct activities that fit the time taxonomy: during development, during maintenance, and during operation. During the development phase, requirements are specified and a design is constructed. This is the groundwork for all future development of this piece of software. If a flaw is not considered or discovered here, it will matriculate into the software program.

During the development, the source code is developed. Without proper testing, the human-crafted code could hold potential flaws. This is also true for the object code. During the maintenance phase the introduced flaws are often attributed to a programmer's failure to understand the system as whole. Flaws fixed during maintenance in this way tend to introduce more flaws. During operation one should consider the introduction of

30

unauthorized modifications during software use. Any change vector from a virus to ordinary users may be able to modify the software and introduce flaws.

Table 13. Flaws by Time of Introduction [LBM93].

Time of Introduction	During Development	Requirement/Specification/Design
		Source Code
		Object Code
	During Maintenance	
	During Operation	

2.3.3.3. Flaws by Location

In this Section the concept of spatiality is introduced. This is a flaw which can be classified according to where in the system it is introduced or found [LBM93]. Table 14 shows how the flaws by location are deconstructed. Most flaws occur in software, but they can also occur in hardware as well. Software spatiality is divided into operating system, support and application. Operating systems are defined as a system which handled memory and processor allocation, process management, device handling, file management, and accounting [LBM93].

This layout is chosen by [LBM93] because it matches the typical layout of known operating systems. The support program category is the other programs which interface with the software including: compilers, editors, databases and other non operating system programs. The application software is a category for programs that have no special system privileges.

31

Table 14. Flaws by Location [LBM93].

Location	Software	Operating System	System Initialization
			Memory Management
			Process Management/Scheduling
			Device Management
			File Management
			Identification/Authentication
			Other/Unknown
		Support	Privileged Utilities
			Unprivileged Utilities
		Application	
	Hardware		

2.4. Intelligence Preparation of the Operational Environment (IPOE)

According to AFPAM 14-118, IPOE is a rigorous analytical methodology focused on providing predictive intelligence to warfighters at the right time for use in planning and executing operations [DAF08]. The steps of the AF IPOE process are outlined in this Section in detail and depicted graphically in Figure 5.

Figure 5. IPOE Cycle [DAF08].

32

IPOE shifts the Intelligence Surveillance and Reconnaissance (ISR) focus from a reactive mode of "discovery" to one where the emphasis is to predict and confirm adversary actions and inject decision-quality information where it can support the commander's objectives [DAF08]. IPOE drives planning, is predictive, actionable, understandable, timely, relevant and tailored. IPOE allows analysts, through observation of adversary actions and behaviors over time, to develop an in-depth cultural, behavioral, and situational understanding of the adversary. The basic steps in the IPOE process remain the same, regardless of the level of military operations. The battlespace, or operational, environment is the first area of focus for the process.

2.4.1. *Step 1: Define the Operational Environment*

The first step of the AF IPOE process focuses effort on defining areas and characteristics of the operational environment that most influence campaign or mission execution. This definition bounds the intelligence problem and highlights, for further analysis, significant characteristics of the operating environment that may influence available COAs or leaders decisions. Defining the operational environment is accomplished by following the below steps [DAF08].

(1) Analyze the mission

(2) Identify the amount of detail that is required and achievable within the time available for IPOE

(3) Identify the limits of the operational area (OA)

(4) Determine the significant characteristics of the OA

(5) Establish the limits of the area of interest (AOI)

(6) Determine the dimensions of the operational environment

(7) Identify intelligence gaps and priorities

(8) Collect the required material and intelligence necessary to complete the remainder of the IPOE process

2.4.2. *Step 2: Describe the Operational Environment Effects*

The second step in the IPOE process determines how the operational environment may affect both adversary and friendly operations. Predicting potential adversary COAs and developing friendly COAs is an important part of the planning process [DAF08]. The operational environment helps or hinders both the adversary and friendly forces when considering these COAs. The primary purpose of this step is determining how the operational environment may affect, positively or negatively, both adversary and friendly operations [DAF08]. This step is broken down into four steps outlined below.

(1) Analyze the physical environment

(2) Analyze the effects of weather

(3) Analyze the human dimension

(4) Describe and depict the effects of the physical environment, weather and the human dimension on friendly and adversary operations

2.4.3. *Step 3: Evaluate the Adversary*

This step involves a detailed study of adversary forces in order to determine the tactics, strengths and weaknesses of the adversary. This evaluation is also used to understand how the enemy typically will behave according to their employed doctrine

34

and tactics. The products of this step may vary depending on the depth of the IPOE analysis being performed, but generally will include identifying enemy centers of gravity (COGs) and critical vulnerabilities (CVs) [DAF08]. The main purpose of this step is to use the evaluation to predict potential COAs for step 4 of the IPOE process. This step is broken down into four phases outlined below.

    (1) Identify and analyze adversary COG/CV

    (2) Create or update threat and other models

    (3) Determine the current adversary situation

    (4) Identify adversary capabilities and vulnerabilities

2.4.4. *Step 4: Determine Adversary Courses of Action*

The purpose of step 4 is to analyze potential adversary COAs and identify develop, and prioritize likely adversary COAs to support friendly COA developments that can be exploited to shape the operational environment and accomplish the friendly mission [DAF08]. This step is broken down into six phases outlined below.

    (1) Explicitly identify assumptions

    (2) Identify the adversary's likely objectives and desired end state

    (3) Develop COAs based on adversary perception of friendly disposition

    (4) Identify the full set of potential COAs available to the adversary

    (5) Identify and nominate targets valuable to the adversary executing probable COAs

    (6) Identify and nominate collection requirements that monitor potential COAs and key operational environment characteristics

35

2.5. Attack Modeling

Attack modeling is a structured design geared towards understanding the complexities of a security incident, such as a computer or network attack. This is the method for portraying system states, vulnerabilities, exploits and attacker actions. These models grant analysts the ability to digest the vast amount of details which comprise a modern computer attack.

2.5.1. *Attack Tree Modeling*

Attack trees are diagrams which model the threat of an attack conducted by an adversary against any type of target. The attack tree modeling concept for computer security threats was developed by Bruce Schneier [Sch99]. His attack tree hierarchy modeling provides a formal, methodical way of describing the security of systems, based on varying attacks. This modeling system offers the ability to decompose, visualize and determine the methods of an attack. Essentially, the attack tree enumerates the methods an attacker uses to reach a goal or objective.

The root node of the modeled tree represents the goal of the attack. Each node in the tree represents a set of sub-goals that must be achieved in order for the top level goal to succeed [LHS03]. Traversing the tree from each leaf of the root node will reveal a potential attack methodology used by the attacker to reach the goal. It is also important to note that an attack tree, or its parts, can become part of a larger attack tree and as such is reusable.

The four major components of an attack tree are: Root, AND, OR and Leaf. The root node is the goal of the attacker, such as gaining unauthorized access to a target

www.manaraa.com

system. The AND nodes and OR nodes operate in a manner similar to AND/OR logic gates. The AND node is only possible if all the children nodes are possible. Essentially all sub-goals must be met in order to achieve the AND node. An OR node is possible if one or more of the children nodes are possible to achieve.

Figure 6 is a simple attack tree of an attacker whose goal is to open a safe. The Figure shows AND and OR nodes, with only AND nodes marked. This example demonstrates how each node becomes a sub-goal, and the children of that node are ways to achieve that sub-goal [Sch99]. The example also includes values assigned to the various leaf nodes, I (impossible) and P (possible). From these values calculations for the security of the goal, or feasibility of attack, can be made. Any value, or metric, can be assigned to the leaf nodes and then propagated up the tree structure in the same manner: easy versus difficult, expensive versus inexpensive, intrusive versus nonintrusive, legal versus illegal, special equipment required versus no special equipment [Sch99]. Any metric that is placed on the tree graph can be used to show if an attack methodology is feasible per the goal and the metric given.

Creation of an attack tree model requires expertise about the attack that is to be modeled. The goal and all sub-goals need to be understood in order to make the model worthwhile. The first step in creating an attack tree is to identify all possible attack goals. Each goal forms a separate tree, although they might share sub trees and nodes. Next identify all attacks against the goals, and add them to the respective tree. Repeat this step until all attacks against these goals are listed. Note that sub-goals of a given goal can form their own respective attack tree model.

37

Building a complete attack tree gives the ability to analyze and research the security decisions that have to be made for attack. The model will show the behaviors of the attacker and how they will act. Therefore a security reaction, response and handling plan for those decisions can be made. The model can be shown graphically or done in simple text as Figure 7 shows. Overall, the attack tree will give an analyst a comprehensive look at the knowledge, methodology and tactics of an attacker through their eyes.
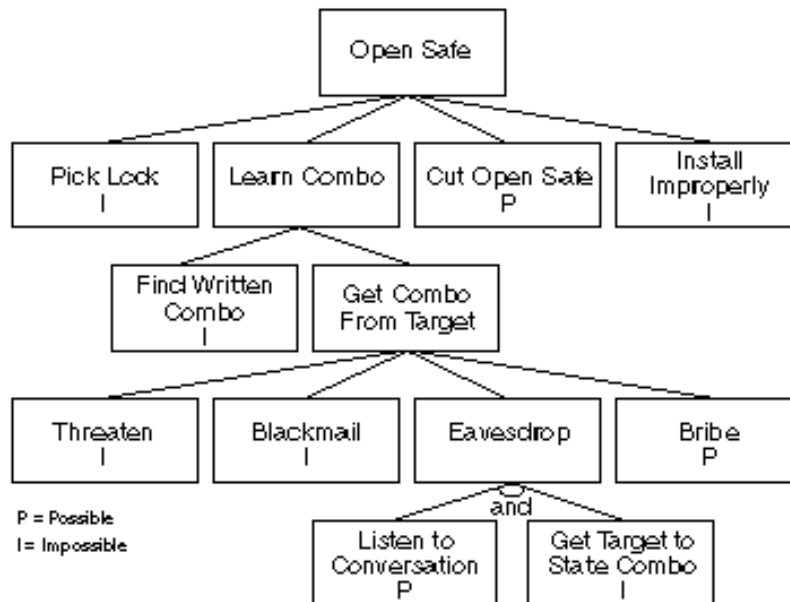
Figure 6. Example Attack Tree of a Physical Safe [Sch99].

2.5.2. *Attack Tree Metrics and Analysis*

Creating an attack tree gives insight into all the possible attack methods an attacker can use to reach their goal. A simple analysis is conducted by starting at each leaf node and traversing up the tree to the root node. Each leaf node traversal will reveal a potential attack methodology. Moving past this simple analysis requires more

38

information be included in the attack tree model. This additional information, or metrics, added to a node is considered weighting the nodes. These metrics can be anything quantifiable such as probability, success, cost, risk, or damage. Typically three or four metrics are used [Ame04]. Too few indicators lead to a flat, one-dimensional understanding of the forces that drive incidents. Too many metrics can confuse the analysis. Ideally, indicators should be orthogonal. This means that the quantifiable influence of one indicator is independent of another. Overall these metrics are considered the capabilities of an attacker.

The metrics associated with a node are defined as capabilities. Capability analysis of a tree model consists of analyzing each of the nodes and determining the value of the metrics to be assigned to the node. Through further analysis, this will determine the likelihood of the attacks. These resource metrics can be considered behavioral indicators because they influence the behavior of adversaries [Ame05]. Capabilities-based analysis of attack trees is rooted on a very simple premise about attackers' behavior: if they want to and they have the capability to do so then they will [Ame05]. In other words, if adversaries exist that have the motivation to harm the system, the resources needed to carry out the exploits and a willingness to accept the potential consequences of their actions, then, at some time, they will carry out an attack on the system.

```
Goal: Read a specific message that has been sent from one Windows
        95 computer to another.
1. Convince sender to reveal message. (OR)
  1.1. Bribe user.
  1.2. Blackmail user.
  1.3. Threaten user.
  1.4. Fool user.
2. Read message when it is being entered into the computer. (OR)
  2.1. Monitor electromagnetic emanations from computer screen.
          (Countermeasure: use a TEMPEST computer.)
  2.2. Visually monitor computer screen.
3. Read message when it is being stored on sender's disk.
          (Countermeasure: use SFS to encrypt hard drive.) (AND)
  3.1. Get access to hard drive. (Countermeasure: put physical locks on
          all doors and windows.)
  3.2. Read a file protected with SFS.
4. Read message when it is being sent from sender to recipient.
          (Countermeasure: use PGP.)  (AND)
  4.1. Intercept message in transit. (Countermeasure: use transport-layer
          encryption program.)
  4.2. Read message encrypted with PGP.
5. Convince recipient to reveal message. (OR)
  5.1. Bribe user.
  5.2. Blackmail user.
  5.3. Threaten user.
  5.4. Fool user.
6. Read message when it is being read. (OR)
  6.1. Monitor electromagnetic emanations from computer screen.
          (Countermeasure: use a TEMPEST computer.)
  6.2. Visually monitor computer screen.
7. Read message when it is being stored on receiver's disk. (OR)
  7.1. Get stored message from user's hard drive after decryption.
          (Countermeasure: use SFS to encrypt hard drive.) (AND)
    7.1.1. Get access to hard drive. (Countermeasure: put physical locks
            on all doors and windows.)
    7.1.2. Read a file protected with SFS.
  7.2. Get stored message from backup tapes after decryption.
8. Get paper printout of message. (Countermeasure: store paper copies
          in safe.) (AND)
  8.1. Get physical access to safe.
  8.2. Open the safe.
```

Figure 7. Text Attack Tree Example [Sch99].

2.5.2.1. Attack Scenarios

An attack scenario is a particular path, or set of paths, through an attack tree that leads from a minimal set of one or more leaf nodes to the root [Ame04]. An attack scenario is the simple analysis as discussed in the previous Section. It is minimal in the

40

sense that, if any of the leaf events are removed from the path, then the root cannot be achieved. Attack scenarios can be found for an entire attack tree. The complete set of attack scenarios for an attack tree shows all of the attacks that are available to an attacker who possesses near infinite resources, capabilities and motivation. [Ame05].

2.5.2.2. Attack Tree pruning

An attack tree contains all possible methods to achieve the root goal. Analyzing each of these attack scenarios of an attack tree for a given attack can lead to time inefficiencies. These wasteful considerations are caused by analyzing attack paths which are outside or not representative of the capabilities of the attacker. An objective, quantitative measurement of the scenarios must be applied to the attack tree in order to alleviate this waste. The determination of the least likely scenarios is the pruning process.

Pruning determines whether or not a particular attack scenario in the tree model should be considered for further analysis. This consideration is determined from an analysis on each attack scenario as to whether a particular type of threat agent has the resources required to reach the goal of the scenario [Ame04]. This act of pruning reduces the attack analysis space. It should be noted that while pruning makes analysis easier, it also can remove viable attack vectors and may potentially leave defenders unprepared.

2.5.3. *Attack Tree Drawbacks*

The attack tree is only a model, and as such it is only as good as the data it is built from. Any data not included, wrongfully identified or incorrectly modeled can have an

41

adverse effect on end results. Also, though a comprehensive attack tree is alluring, it is not always the best choice. The complexity and upkeep of the data has the potential to become overwhelming. The more thorough an attack tree, the more complex it becomes which, limits its illustrative nature [Kar05]. There is a delicate balance of complexity and usability that can hinder the models effectiveness.

2.6. Related work

This Section covers related work in the cyber battle damage field of study. The following research topics focus on damage and impact assessment.

2.6.1. *The Horony Damage Assessment Model*

In 1999, Captain Mark D. Horony developed an Information System (IS) incident Damage Assessment Model (DAM) tailored to the needs of the IS manager's organization [Hor99]. This model is shown graphically in Figure 8. The model was designed to be part of a IS manager's toolkit to ensure a full and accurate damage assessment has been accomplished for security incident response [Hor99]. The processes of a business are the foundation for the DAM. The manager of an IS must define these processes and how they relate to the information systems of their control.

Horony based the incident assessment process on eight primary factors listed below. Each of these is further expanded into sub-factors which better define and explain the primary factors [Hor99].

(1) Recovery

(2) Education/Training

(3) Business Expenses

(4) Productivity

(5) Data

(6) Lost Revenue

(7) Reputation

(8) Human Life
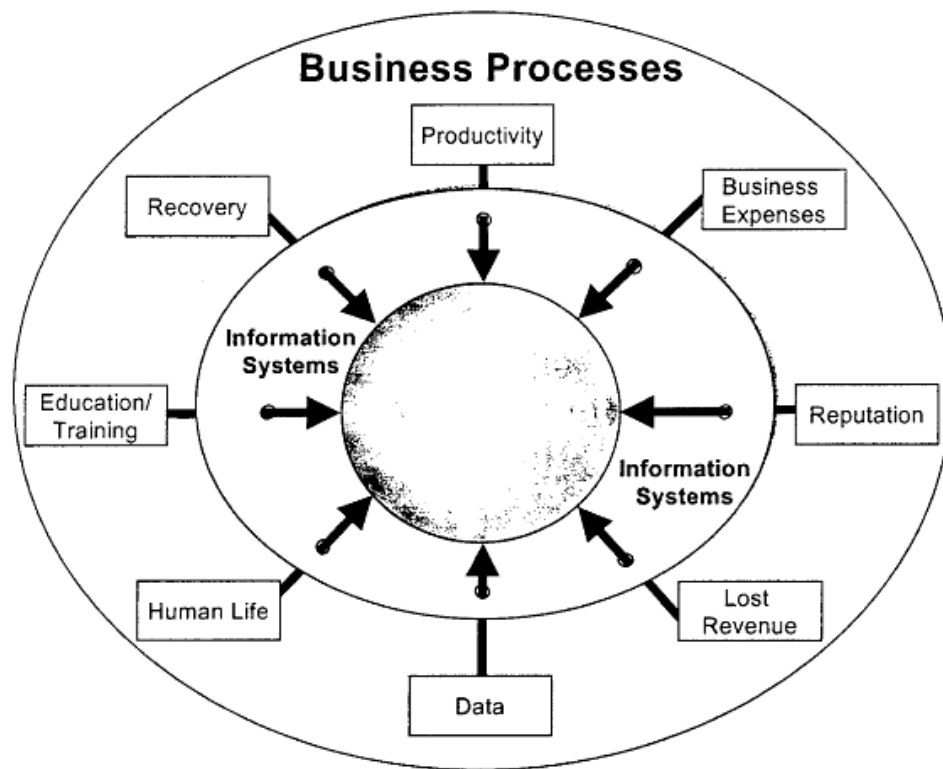
Figure 8. Horony IS DAM [Hor99].

2.6.1.1. Recovery

Recovery is the process that a system administrator must take to restore an IS to

the most current state prior to an incident [Hor99]. The operations to support recovery

43

include backups, manual data input, and repair of damaged software or hardware. Recovery is a factor which includes all issues that must be accomplished to restore the IS to the previous working state [Hor99]. The sub-factors for Recovery defined by Horony are listed below.

(1) Investigation

(2) Restore

(3) Software /Hardware

(4) Consultants/Contractors

(5) Accounts

2.6.1.2. Education/Training

Education and training is a necessary factor when considering the various business processes and information systems a user must understand [Hor99]. During the course of an incident investigation the need for additional education and training within the organization may become evident. System administrators and information security personnel may not have the necessary skills to perform a thorough investigation [Hor99]. As new procedures and processes are implemented in an organization, so too does the need to implement training and education of the systems. A goal of this training is to reduce the overall chance of future incidents. The sub-factors of Education and Training are listed below.

(1) System Administrator/Information Security Personnel

(2) Employee Computer and Information Security

44

2.6.1.3. Business Expenses

Business Expenses are the direct fees or costs that result from outages, which affect customers and other businesses [Hor99]. An IS incident impacts the business process of an organization, and as a result causes cost factors from IS outages. The two sub-factors of Business Expenses are listed below.

(1) Customer Service

(2) Business to Business

2.6.1.4. Productivity

When an information systems' performance is degraded, the productivity of an organization will be impacted [Hor99]. An organization's business processes must be determined and understood in order to evaluate how the organizations productivity will be impacted during and IS incident. Productivity has three sub-factors listed below.

(1) Mission Impact

(2) Downtime

(3) Communication

2.6.1.5. Data

Data lost as a result of an information incident must be evaluated for the ability and cost of recovery [Hor99]. The impact of how an organization handles data, from storage to usage, must be evaluated as a result from an IS incident. This impact must be

45

understood in order to lessen the impact of future events. The sub-factors of Data are listed below.

(1) Restoring

(2) Re-Entering

(3) Unrecoverable Data

(4) Proprietary Data

2.6.1.6. Lost Revenue

If a system is damaged it is necessary to evaluate how it will impact the revenue generating process of the organization [Hor99]. There are two sub-factors to consider for Lost Revenue listed below.

(1) Lost Sales

(2) Lost Customers

2.6.1.7. Reputation

Reputation is an overarching opinion of a company. This factor can be impacted from an IS incident. This can in turn impact customers, employees, revenue and other aspects of an organization [Hor99]. While difficult to quantify, the incident assessment must include considerations for the reputation of the impacted organization. Reputation has two sub-factors listed below.

(1) Consumer/Public Confidence

(2) Quality Employees

46

2.6.1.8. Human Life

The consideration toward the human life factor in an organization is very important. An IS may support life saving assets or missions. The morale and personal life of information security personnel who respond to incidents must also be considered. The basic fact to factor is that the quality and standards of human life depend on IS and may be impacted by an organization's security incident. The sub-factors of human life are listed below.

(1) Loss of Life

(2) High work load of Emergency Response Team members

2.6.2. *Fortson Case Study of Damage Assessment*

In 2007 Captain Larry W. Fortson established a conceptual operations-focused methodological framework that facilitates effective defensive cyber damage and mission impact assessment and reporting following cyber-based information incidents [For07]. The research focused on the relationship of damage assessment and mission impact. The research used historical analysis and case study methodology for data collection through literature review, examination of existing case study research and interviews with Air Force members and civilian personnel employed as experts in cyber damage and mission impact assessment of Air Force networks [For07].

The research found that damage assessment is conducted in a disjointed manner and in many cases is limited to technical or economic reporting with no real assessment of damage in terms of value loss [For07]. Damage assessment must consist of both a

47

technical assessment and a damage assessment that measures a cyber value loss. The prevailing theme throughout the research was the need to refocus the fixation on technology to allow a more comprehensive understanding mission impact assessment [For07].

Fortson states that damage from a successful cyber attack may be measured effectively only if the value of the asset is known before the incident. Fortson's research also found that Air Force organizations are not looking at the right assets for damage assessment, due largely to a failure to recognize what assets support the mission [For07]. An organization must be able to identify the correct assets which support their mission. They must also be able to articulate a value of the asset in order to provide substantial damage assessment. It is through an organization's ability to identify and value an asset before an incident occurs that cyber damage assessment is possible.

Fortson introduced the Defensive Cyber Damage and Mission Impact Assessment (CDA-D/MIA) as an improved methodology over existing mechanisms. The CDA-D/MIA methodology is intended to assess damage and mission impact in the organization where the incident occurred and provide rapid reporting of the assessment results to the local decision maker, NETOPS command and control structure, and the appropriate interested report consumers [For07].

In his framework, Fortson shows how an Incident Response Agent conducts a technical damage assessment which is a separate report from the mission impact assessment. This data flow of the technical damage assessment is depicted in Figure 9.

48
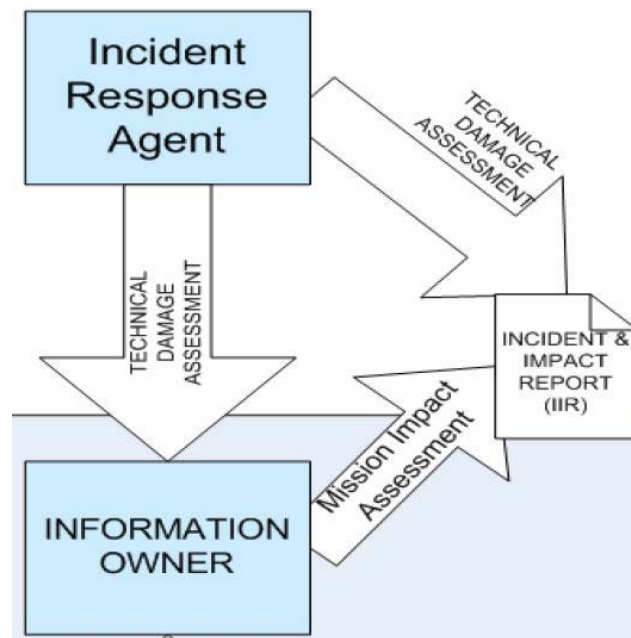
Figure 9. CDA-D/MIA Incident and Impact Reporting [For07].

 Based on these technical assessments the information owner can begin to
estimate damage by comparing the amount of critical information asset exposure to the
threat. The information owner and the incident response agent work together to determine
if the threat has resulted in any damage [For07].

Fortson's research showed that mission impact assessment is too focused on the
technical damage assessment and lacks a vehicle for comprehensive understanding of
mission impact due to digital asset degradation. Fortson does not design a methodology
for technical damage assessment, rather a framework for mission incident and impact
assessment. The purpose of network defense is to protect the information assets on the
network and Fortson's framework seeks to give the defenders the avenue to communicate
how those assets are impacted by a security incident.

49

2.6.1. *Investigating Computer Attacks Using Attack Trees*

In 2007 Poolsapassit and Ray introduce an attack-tree based filtering algorithm that eliminates information from a log file that is not related to an attack being investigated and extracts evidence corresponding to a particular source's role in the attack [PoR07]. The research describes an automated attack-tree-based approach for filtering irrelevant information from system log files and conducting systematic investigations of computer attacks.

The research defines an "augmented attack tree," which extends the basic attack tree by associating each branch of the tree with a sequence of malicious operations that could have contributed to the attack [PoR07]. Figure 10 shows how an augmented attack tree may be used to support a forensic investigation.
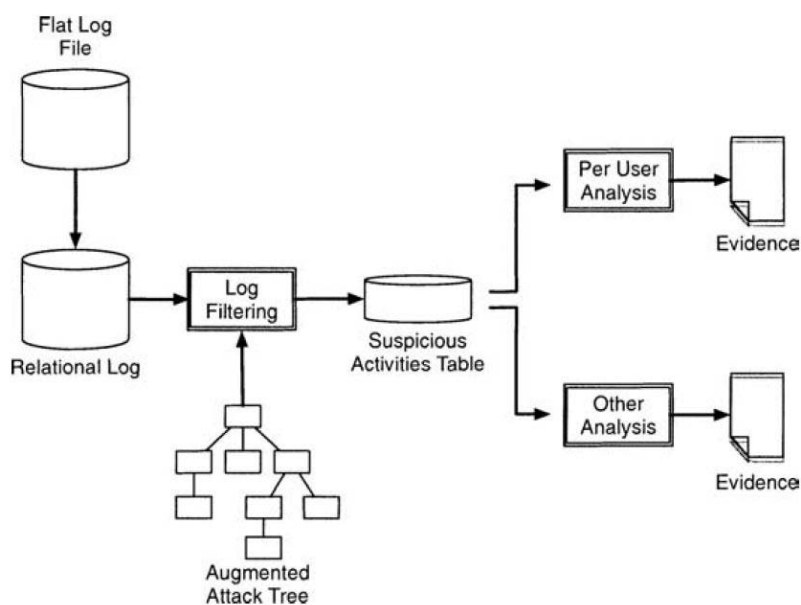
Figure 10. Log File Investigation Process [PoR07].

50

To conduct a forensic investigation the augmented attack tree is first used to generate the set of incidents corresponding to all the attack signatures for the system [PoR07]. Each edge in the attack tree specifies an attack signature. Each attack signature is a collection of several incidents. Then a log file filtering algorithm sequentially executes queries to extract suspicious activities, from non-suspicious ones, from an original log file from the attack. The algorithm starts at the root node of the attack tree. It traverses every edge incident to the root node. For each step in the attack signature, the algorithm searches the log file for matching operations. Then all the subtrees under the node are explored recursively. After all the subtrees under the root node or any intermediate node have been explored, the algorithm marks an edge if it finds evidence that shows that all the steps in the attack signature have been executed.

The next step in the process is to process the data produced by the log file filtering algorithm for candidate sources of the attack [PoR07]. This is accomplished by sorting the data by source to produce the list of candidate sources. The output of the sorting process is the identity of the source being investigated. Therefore, the algorithm should be used very carefully as it only provides evidence of activities that were possibly involved in an attack. An investigator may use the identified suspected records with the applied corresponding exploit labels to map the evidence back to exploits in the attack tree.

This research shows how an attack tree can be used to filter and extract attack evidence information from a large set of logged data on an attacked system.

51

III. Methodology

This chapter presents the methodology used to create the CAMEL, the DCBDA process and the steps used for the experimentation of this thesis. Section 3.1 covers an overview to the methodology. Section 3.2 details the CAMEL and CAMAT creation approach. Section 3.3 covers the DCBDA preparation phase approach. Section 3.4 discuses the DCBDA forensic analysis. Section 3.5 details the experimental design. Section 3.6 summarizes the chapter.

3.1.   Methodology Overview

The primary goal of this research is to create an analytically rigorous, defensively focused cyber battle damage assessment process which utilizes the detailed understanding of the attack methodology. The ability to analyze an attack at any level of progress is vital to a network's ability to survive and operate in a hostile environment. Knowing from the studied nature of the sophisticated attacker that a cyber attack is likely to have some degree of success, it is crucial for an organization to possess the ability to determine if a cyber attack has occurred.

The cardinal focus of the proposed DCBDA process is to utilize the intelligent comprehensive knowledge of the cyber attack methodology captured in the CAMEL and organized in a CAMAT to identify cyber attack forensic markers for an active attack incident. Before the DCBDA process can be used, the CAMEL and CAMAT must be created. Once created, these products are continuously updated with new data by the owning organization.

The CAMEL contains the best known set of all collected data concerning the actions in cyber attacks. An overview of the CAMEL creation process and CAMEL data relationships are shown in Figure 11. The data in CAMEL includes attack actions, action metrics, attack tools or methods, forensic evidence markers and forensic tools or methods associated to the evidence. Possible sources for CAMEL data are forensic analysis white papers, ethical hacker documentation, real world reports, and personal experiences. The CAMAT is the attack tree model built from the comprehensive data in CAMEL. As discussed in Chapter 2, an attack tree provides a detailed model in which to gain an understanding of the attack methodology that may be used against a target system.
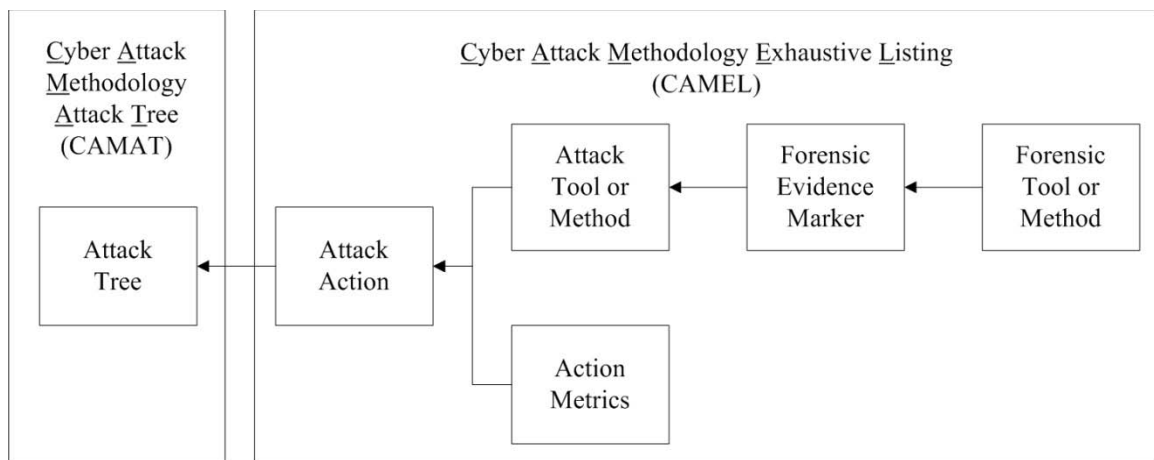
Figure 11. CAMEL and CAMAT Creation Process.

It is an important goal for this research to create the CAMAT model. The CAMAT allows for detailed modeling of all key actions an attacker takes during an attack. This in turn allows for attack COA study and understanding. The tailored enumeration of attacker actions from the CAMAT pinpoints the exact files, settings, logs

53

and possibly the bits that an attacker modifies during an attack. By building an exhaustive list of attack actions, their related effects and modeling those actions in an attack tree, robust damage assessment of a cyber attack is possible.

Once the CAMEL and CAMAT exist, the DCBDA process can be utilized for assessment. The DCBDA process in total is depicted in Figure 12. This figure shows the two phases of the DCBDA process.
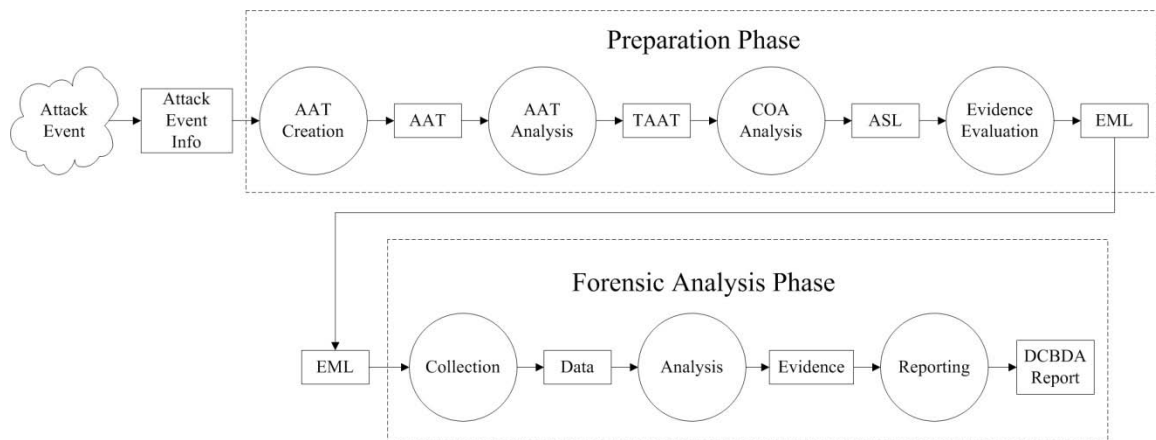
Figure 12. DCBDA Process.

The preparation phase begins with attack information derived from an attack event. An example of this attack information could be the knowledge that a malicious email attachment was opened. This information is then applied to the creation of the Assessment Attack Tree (AAT) model, which is an event assessment specific attack tree built from a subset of the CAMAT. The AAT is then analyzed and tailored to the attack under assessment. This step creates the Tailored AAT (TAAT). The event TAAT is then analyzed for likely attack scenarios, or COAs, which are ordered and listed in the Attack

54

Scenario List (ASL). The ASL is then evaluated for evidence markers and associated tools or methods to retrieve those markers. The evidence, tools or methods and the actions to which they are associated are listed in the Evidence Marker List (EML). The EML is the final product of the preparation phase of the DCBDA process.

The last phase of the DCBDA process is to conduct digital forensic analysis on the target system utilizing the EML as a collection guide. The digital forensic process varies by organization, but for the purposes of this research the steps outlined in Section 2.2.4, collection, examination, analysis and reporting will be used.

The forensic analysis starts with the evidence collection action on a target system. The collection action uses the preparatory information, the list of software tools or methods for evidence gathering, from the previous phase of the DCBDA process. These tools, once executed, will harvest raw data for evidence analysis if found on the target.

This evidence analysis will determine if an attack occurred and the details surrounding the attack such as damage and enemy COA identification. These findings will answer the key questions of cyber attack. It is important that the potential damage be validated as much as possible from the analysis of the forensic data. This detailed analysis ensures the mission essential technical and operational impact reports, derived from DCBDA assessment, are as accurate as possible.

The final action in the forensic analysis and the DCBDA process is a report. The report lists, among other things, the collected data from the assessment. This data will include the list of evidence found from collection and the associated attacker action which caused the evidence to be generated. Potential attack vectors, system weaknesses,

root causes and incident categories should also be included in this report. The report will detail the findings of the analysis and finalize the DCBDA process.

3.2. CAMEL and CAMAT Creation Approach

This Section covers the creation of the CAMEL and CAMAT as overviewed in Section 3.1 and Figure 10. The CAMEL is the comprehensive knowledge of all known cyber attack methodologies comprised of attack actions, attack methods to produce the actions, metrics of the attack actions, forensic evidence markers from the actions, and forensic tools to retrieve the evidence markers. CAMAT is the attack tree model of the collected data in the CAMEL. It should be noted that only one CAMEL and CAMAT product is needed per organizational implementation. The steps for CAMEL creation as covered in this Section are outlined in Appendix G and listed below.

1. Collect attack actions
2. Analyze attack actions for attack methods
3. Analyze attack actions and methods for forensic evidence
4. Analyze evidence for forensic tool(s) or method(s) for collection
5. Analyze metrics for attack actions
6. Model the CAMEL as CAMAT

3.2.1. *CAMEL Attack Action Data Collection*

In order to build the CAMEL, data concerning known attacker methodology must be gathered, normalized and input into a list. The primary data point in the CAMEL is the collection of attacker actions which comprise the entire cyber attack methodology.

The attack action data in CAMEL should include all known COAs and TTPs for all known cyber attacks. The sources for this data should be varied, accurate and as robust as possible. Possible examples include: forensic analysis white papers, ethical hacker documentation, real world reports, and personal experiences. Regardless of the source, the overall goal of this aggregation is to detail every cyber attack in its entirety, as part of the whole cyber attack methodology, as best as possible and with the most detail.

In order for the CAMAT model to be useful for an assessment, the data in which it is modeled from must be complete and accurate. In order to build the CAMEL with relevant useful data, the CAMEL Attack Action Data Form shown in Table 15 should be used to standardize input collection. When collecting data for a given action, it is important to consider the actions taken before or after the action being analyzed. These actions may have an impact on the data being collected for a given action.

Table 15. CAMEL Attack Action Data Form.

Field	Description	Example
Attack Action	This is the identifier given to a specific action	Heap Overflow
Parent Action(s)	These are the parent nodes of this action	Gain access to web server
Child Actions(s)	These are the children nodes of this action	Malformed data
Attack Vector	This is the means which allowed the action to exist	Configuration error
Attack Results/Goal	This a description of the actions intended results on the target	User Access

57

The attack action form will ensure a standardized schema is followed for the gathering of necessary information needed for the DCBDA process. Note that this form should be modified to fit a particular organization's requirements for data aggregation.

3.2.2. *CAMEL Attack Method Analysis*

After the attack actions have been identified, the next set of data to input into the CAMEL is the enumeration of the attack methods which produce the attack actions. This data is collected from an analysis of each attack action in CAMEL. The analysis must determine all possible methods which are able to produce the action. The method data discovered from the analysis extends the information in CAMEL about an attack action.

Table 16 shows the attack method data collected in the CAMEL which should include the name of the attack method, the attack action the method creates, a description of what the method achieves and a source to retrieve the method for further research. The attack method data allows for a detailed understanding of how the action is created.

Table 16. CAMEL Attack Method Data.

Field	Description	Example
Attack Method	This is a known set of the tools or procedures that create this action	User input into web form
Attack Action	This is the attack action the method creates	Command line access to web server
Attack Method Description	This is a description of the attack method	Special characters in web form not properly handled
Attack Method Source	This is a source for the attack method	Forensic whitepaper

58

When analyzing the scope of the attack action methods of a cyber attacker, the analysis should not confine span and definition.   The fact of the matter is when considering what methods a cyber attacker can use, there is practically no limit.   Cohen states that there is no fundamental distinction between information that can be used as data, and information that can be used as program [Coh87].   Essentially data is code and code is data.   This consideration must be applied when creating input for the CAMEL.

3.2.3.   *CAMEL Evidence Analysis*

This part of CAMEL creation conducts analysis on the attack methods and actions to identify the forensic evidence markers which act as a digital fingerprint of the attack. This effort requires a significant amount of research and analysis of the attack data.   The attack actions and methods must be thoroughly analyzed and every detail of its effects recorded.   The data collection from this analysis must enumerate every trace of evidence on a system that can lead to the positive identification of the attack method or action occurrence on a system.

A measure of confidence analysis must be applied to the identified evidence.   The data included in CAMEL from this analysis is shown in Table 17.   Bearing in mind that there are no universal processes or scientific underpinnings in the methods used to recover or interpret digital information, a level of effort must be applied to give measure to the certainty of the forensic analysis action [GiM02]. The confidence analysis can be a qualitative value associated to the ability of the evidence marker to properly identify the associated attack.   This will allow a DCBDA analyst to measure the relative ability or trust placed in the identified evidence for a given attack scenario.

59

Table 17. CAMEL Evidence Data.

Field	Description	Example
Evidence Name	Name given to the evidence	Hidden file registry setting
Attack Action/Method	This is the attack action or method the evidence identifies	Folder hidden option
Forensic Markers	These are the details of the evidence	HKEY_CURRENT_USER \Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced
Evidence Confidence (Low, Med, High)	This is a metric given for the confidence of the evidence data	High
Evidence Source	This is a source for further information regarding the evidence	Mircosoft.com

3.2.4. *CAMEL Forensic Tools and Method Analysis*

The next step of CAMEL creation is to identify the forensic tool or method to retrieve the identified forensic evidence markers. The specific data to be captured is shown in Table 18. Each evidence marker is evaluated to associate a forensic tool or method with the ability to harvest it from a system. These tools and methods must be tested and have the proven ability to retrieve the markers they are correlated with. A measure of confidence and a source must also be included for this analysis. These identified tool and methods will be used in the DCBDA forensic analysis phase to determine if an attack has occurred.

60

Table 18. CAMEL Forensic Tools and Method Data.

Field	Description	Example
Forensic Marker	These are the details of the evidence	A registry setting
Forensic Tools/Methods for Collection	This is the tool or method to collect the evidence	Regedit.exe
Tool/Method Confidence (Low, Med, High)	This is a metric given to the tool or method for a confidence value	High
Tool/Method Source	This is a source for further information regarding the tool /method	Microsoft.com

3.2.5. *CAMEL Attack Action Metrics Analysis*

In order for the data in CAMEL to support detailed attack analysis, metrics must be assigned to each action. These metrics allow the attack actions to be compared and analyzed for a particular attack being studied. These metrics can take on any form relevant to the organizational entity which uses the CAMEL such as risk, cost and impact. For this thesis research two values will given to each attack action. These values are risk of attack discovery and attack impact. An organization using CAMEL and DCBDA operationally should consider a higher level of emphasis being placed on attack metrics such using industry standard guidelines for metric creation and management [NIS08].

3.2.6. *CAMAT Model*

Once a best known complete set of data for the cyber attack methodology is collected, an attack tree modeling the cyber attack data can be created. This model is referred to as the CAMAT, as that it is the graphical representation of the cyber attack in its entirety. The first step in attack tree creation is identifying the top level or root goal. This step is the effect the attacker wishes to achieve. All other actions taken for this root goal are considered subordinate and supporting to this goal. All of the data collected for the given root node should be used in creation of the tree to ensure completeness to model all aspects of the attack.

From the identified root goal, immediate children nodes should be determined. A child node is an action which creates or supports its parent node. The relationship between the child nodes does not have to be binary to ensure completeness, but the union of the child nodes must cover every possible type of attack for the vulnerability listed in the parent node [Edg07]. This process of identification of new children nodes is now completed for the previously identified children of the root goal. This process is continued for all nodes listed until the complete set of aggregated data has been added to the tree with the appropriate parent child relationship maintained and modeled. Each node must also be modeled as either a AND, OR or leaf node designation as discussed in Section 2.5.1. Each node must also allow for a reference to the verbose data in CAMEL which it was built from. Below is a pseudo code algorithm which shows how to build an attack tree.

Algorithm 1 Attack Tree Model Creation Algorithm

Root Node: The attack goal
Parent Node: A goal which has supporting goals required for fulfillment
Child Node: A goal which is subordinate and creates or supports its parent goal

Build root node
Set root node as parent
While a parent has a child
      Build children node(s) of parent
     If a child is a parent
         Set child as a parent
End While

3.2.7. *Assumptions*

This effort assumes that the evidence evaluation process results in meaningful data output which leads to the identification of a marker for collection. An assumption for tool selection must also be noted here. The manner in which what tools are available for use during collection is outside the scope of this research. Ideally metrics should be established that help determine the extent that a software or hardware tool performs a particular forensic function, and the associated error rate with that process [GiM02]. From this tool analysis and selection process a refined tool repository would be available for the evidence collection tool selection for the DCBDA process.

3.3.   DCBDA Preparation Phase Approach

This approach covers the steps necessary to complete the DCBDA preparatory actions as shown in Figure 13. The preparatory actions of the battle damage assessment process are vital to understand the technical impact of a cyber attack. The purpose of the preparation phase is to utilize the intelligent comprehensive knowledge captured in

63

CAMEL and modeled in CAMAT to determine the forensic evidence and collection methods for the forensic analysis. There are four steps of the DCBDA preparation phase:

1. AAT Creation

2. AAT Analysis and TAAT Creation

3. COA Analysis and ASL Creation

4. Evidence Evaluation and EML Creation
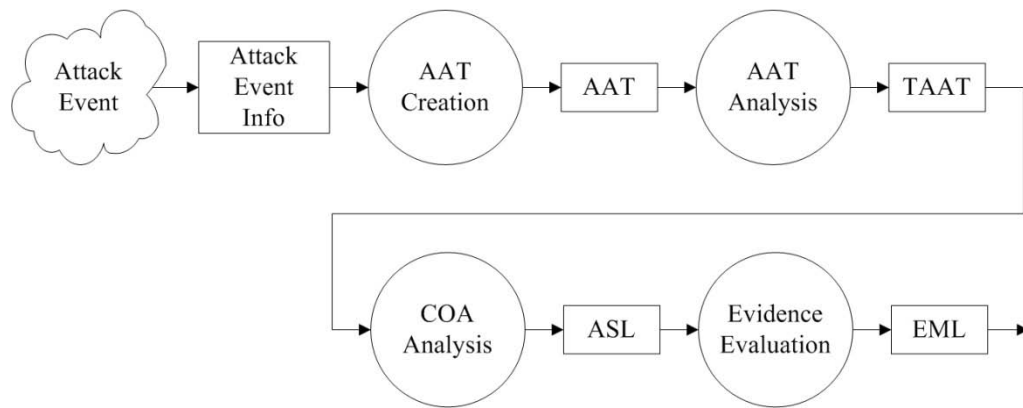
Figure 13. DCBDA Preparation Phase.

The first step in the process is to create the attack tree for the attack event under assessment. This tree is the AAT, which is created from a subset of the CAMAT. This step starts by gathering the relevant information concerning the attack being analyzed. This information must include the specific attack actions which are deemed the purposes of the cyber attack. Once the attack actions are known, the CAMEL/CAMAT information can be used to create the AAT for the attack being assessed.

64

The second step analyzes the AAT and tailors the model to the attack being assessed. This is the capability analysis which is completed to prune and prioritize the possible actions of the attack under assessment. This step creates the TAAT.

The third step selects and orders likely attack scenarios from the TAAT. Attack scenarios are the collective set of possible COAs followed during the attack. Once these attack scenarios have been listed in an ASL, the evidence of the attack can be identified.

The final step of the preparation phase focuses on evidence evaluation based on the information in the ASL. Knowing the enemy's detailed cyber attack COAs enables the identification of important evidence markers from the CAMEL to evaluate if a cyber attack, or in other words an executed adversary COA, took place on a target system. The identified markers must then be evaluated to associate the trusted forensic software programs or methods listed in CAMEL that can retrieve the evidence markers from assessment target. The evidence marker, the action it identifies and the forensic collection method(s) comprise the Evidence Marker List (EML) product of this last step in the preparation phase of the DCBDA process.

3.3.1. *AAT Creation*

The first step in the preparation phase of the DCBDA process is to create the AAT. The steps of this process are shown graphically in Figure 14 and are:

1. Determine the root goal of the attack

2. Search CAMEL/CAMAT for the root goal

3. Transfer data of root goal and all subordinate children actions from CAMEL/CAMAT to create the AAT

www.manaraa.com
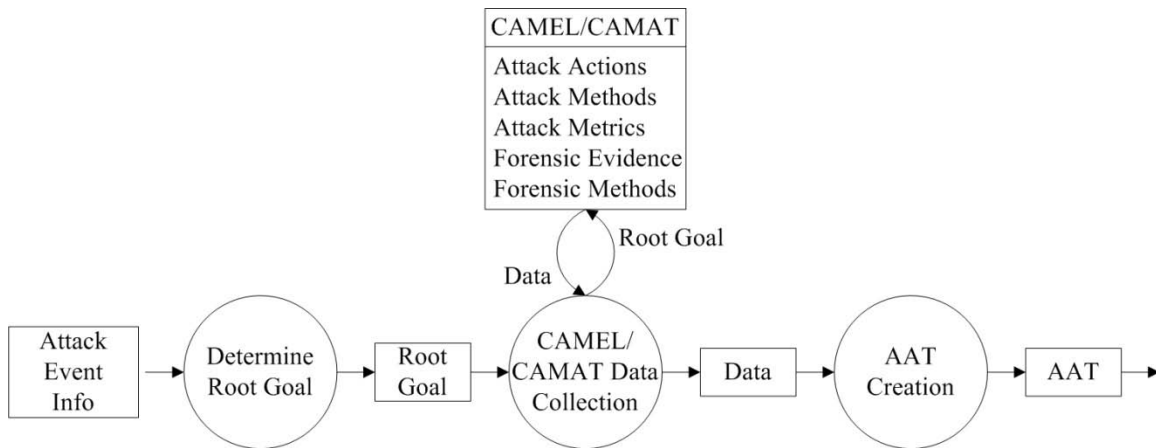
Figure 14. DCBDA AAT Creation.

The first step of AAT creation is to determine the root goal of the attack from the gathered relevant information concerning the attack event being analyzed and assessed. This information must include the specific attack action which is deemed the purpose of the cyber attack. This is the root goal for the assessment. Once the root attack action is known, the information in the CAMEL/CAMAT can be used to create the AAT.

The next step searches the CAMEL/CAMAT for the root goal identified in the previous step. If the goal is not specifically found, the assessment must decide the most reasonable parent goal that is of the same category of the attack. An example of this would be an unknown method for gaining access to a computer. The root goal for this example would be gaining access.

The final step of AAT creation creates the AAT from the data in the CAMEL/CAMAT. When the root goal is found in CAMEL/CAMAT the data is copied and transferred starting with the root goal, to include all children and their associated children. This data retains the parent child relationships, thus remains an attack tree. This data is now the AAT and can be used for attack analysis.

66

3.3.2. *AAT Analysis and TAAT Creation*

Once the AAT has been created, it is time to apply a level of analysis on the segment of data which has been aggregated and modeled for the DCBDA. This analysis process is depicted in Figure 15. While the goal of the CAMEL is to be as complete as possible, the end goal for the AAT analysis is to prune the total set of attacker COAs to a manageable, applicable and believable set. This will be captured in the TAAT. It is an intuitive approach to this analysis that realizes an attacker is much less likely to try every single attack method, rather than just a few of the most probable or applicable. The two parts to this step of the preparation phase of DCBDA are:

1. Capability Determination

2. Capability Analysis

The first step in AAT analysis is to apply capability determination to the attack actions represented by the values of the associated metrics. This analysis uses the collected attack event information to update and/or apply new metrics to the attack actions in the AAT. Essentially this step creates an assessment AAT which contains updated data to reflect the particular attack being assessed. The same level of rigor used to create the metrics for CAMEL should be applied here. For the purposes of this thesis, the metrics will not be updated or modified. Once the capability determination is complete, capability analysis can be accomplished.

Capability analysis, the second part of this step, determines which actions should not be considered for inclusion in the DCBDA TAAT. This analysis uses the attack action metrics to determine if an attack action should be removed from the AAT and the

67

attack assessment. This determination can be decided from a known attacker capability, attack event data or an arbitrary decision by the DCBDA analyst. This analysis process is known as pruning, as the resulting attack tree will be smaller than the original. It should be noted that the pruning process can remove plausible potential attack vectors from the attack tree. The product of this step will be a TAAT which is an attack tree with the realizable set of attack scenarios for the given attack event being assessed.
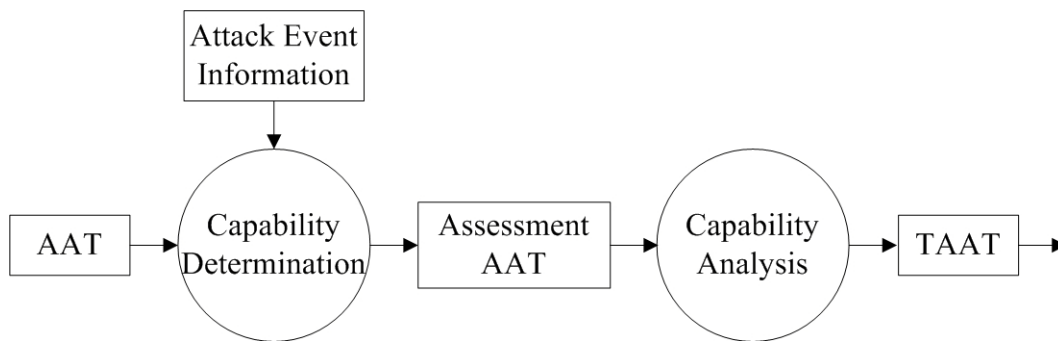
Figure 15. DCBDA AAT Analysis and TAAT Creation.

3.3.3. *COA Analysis and ASL Creation*

This step of the preparatory actions for DCBDA is COA analysis and ASL creation. This step is outlined in Figure 16. Recall from Section 2.5.2.1 that an attack scenario is a particular path through an attack tree that leads from a minimal set of one or more leaf nodes to the root. At this point the TAAT will have numerous possible attacks to achieve the root goal. In order to maximize the efficiency and accuracy of the damage assessment, attack scenarios should be identified, listed and ordered according from the most to the least likely. This list is referred to as the ASL. This intelligence product

68

provides a decision maker with a threat assessment based on an analysis of the full range of adversary capabilities and a prediction of the adversary's likely intention [DOD07]. The ASL is based on the given collected attack data which has been modeled into information, analyzed by capability and coalesced into a tailored attack tree model.
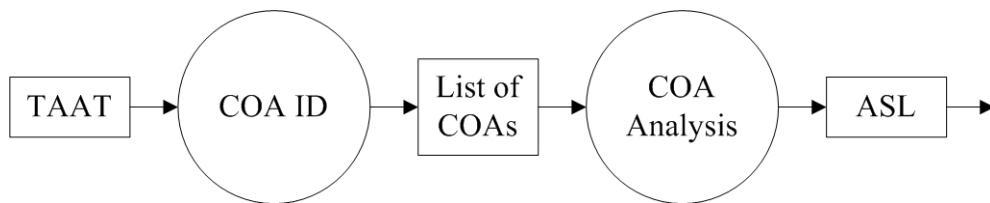
Figure 16. DCBDA Attack Scenario Analysis and ASL Creation.

The priority of attack scenarios can be decided from the combined results of the TAAT capability analysis or an arbitrary determination from the DCBDA analyst. Depending on the size of the attack tree being analyzed it is very useful for time considerations of an assessment to develop a prioritized list of attack scenarios. This allows the DCBDA forensic analysis to place the analysis of probable attack scenarios ahead of unlikely attack scenarios. This ordering will also enable a threshold decision to be made when considering what forensic evidence collection tools to use on the target system. An example format for this listing is shown in Table 19. The ASL is the product of this stage of the DCBDA process.

www.manaraa.com

Table 19. DCBDA ASL.

Attack Scenario Data	
Scenario Name	
Attack Action Listing	

3.3.4. *Evidence Evaluation and EML Creation*

The final step in the preparation phase of the DCBDA process is evidence evaluation and EML creation. The actions of this step are outlined in Figure 17. The goal of evidence evaluation is to use the listed attack scenarios in the ASL to identify the forensic evidence markers and the forensic tools necessary to determine if the actions took place on a system. This process takes the relevant data in the CAMEL and transfers it into the EML.
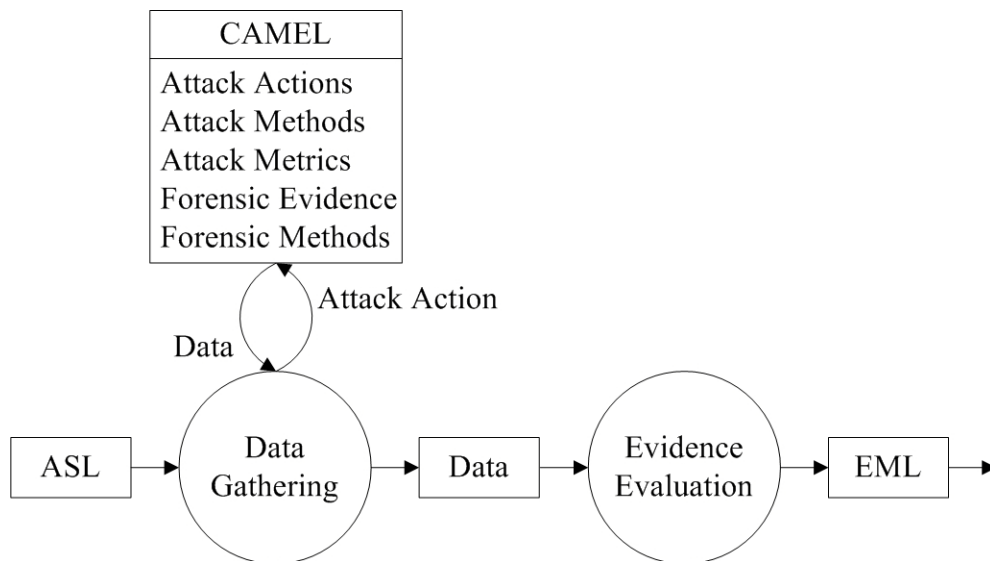
Figure 17. DCBDA Evidence Evaluation and EML Creation.

The EML is the specific evidence and tools needed to identify the attack being assessed. The evaluation of the evidence should take into account the operating environment of the analysis in determining the appropriate tools to identify. The EML is the product of the evidence evaluation and is the final product of the preparation phase of the DCBDA process. The EML is next used for the forensic analysis process.

Table 20 holds the format which is used as a record in the EML product. In total the EML is comprised of a record, or entry, for each action in the attack scenario which has information regarding the name of the action, the evidence identifiers which identify the action, and the associated tool which retrieves the identifiers.

Table 20. DCBDA EML Data.

Field	Description	Example
Evidence Name	This is the name given to the evidence	Hidden file registry setting
Name of Action	This is the identifier given to a specific action	Heap Overflow
Forensic Marker	These are the details of the evidence	A registry setting
Forensic Tools/Method	This is the tool or method to collect the evidence	Regedit.exe

71

3.3.5. *Assumptions*

The exact makeup of the AAT will change depending on many factors. The target system can have a large impact on the makeup of the attack tree. A particular OS may have a very different attack vector than an OS version with only a service pack in difference. For the purposes of this research, the focus will remain on one target system. This will allow the data in the DCBDA process to be demonstrated, and therefore have the ability to be expanded or created new for different targets.

3.4. DCBDA Forensic Analysis Approach

This phase of the DCBDA research effort is the collection, analysis and reporting for the forensic analysis. A graphical representation of this phase is shown in Figure 18. This phase acts on the culmination of data aggregation, modeling, evaluation and analysis of the DCBDA preparation phase. This step also takes the DBCDA process from the intelligence preparation process into active defense assessment. The goal of DCBDA forensic analysis is the recovering, analyzing and presenting of evidence of a potential attack on a target system. This research will follow the analysis guidelines found in Section 2.2.4 which is further detailed in the National Institute of Standards and Technology: Guide to Integrating Forensic Techniques into Incident Response [NIS06].
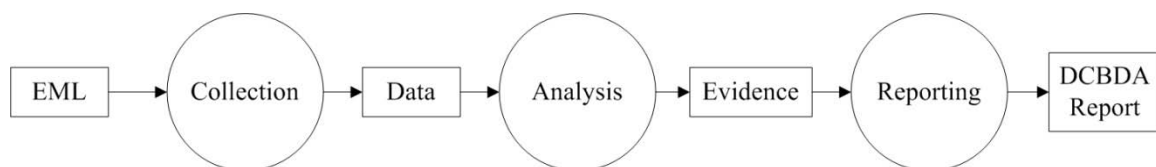
Figure 18. DCBDA Forensic Analysis.

72

3.4.1. *Evidence Collection*

This step of the DCBDA is the execution of the tools identified in the EML. The listed tools, or forensic COAs, will collect the evidence needed for the attack analysis. It is important that a standardized process be followed when running these tools. The exact process of collection will vary according to organizational procedures, tools, environment and many other factors. There are many examples of guidelines which may be used for the collection of evidence during a forensic analysis [DOJ04] [NIS06].

3.4.2. *Evidence Analysis*

With the identified evidence for the attack collected, it is time to begin analysis of the attack. This process is the investigation into the meaning of what evidence was found on the target system. Recall from Section 2.2.4.3 that the objective of analysis is to derive useful information which forms conclusions to the questions that were the impetus for performing the collection and examination [NIS06]. There are several methods of analysis that may be conducted depending on the type of attack and the data collected. Some examples of analysis that may be performed include timeframe, data hiding, application and file, and ownership and possession [DoJ04].

The collected data must be processed into usable information. This process examines the set of data collected from the target system and locates the evidence within. Essentially the examination gleans the pertinent useful data from the total set of information harvested from the forensic tools. This data is then analyzed to determine if an attack occurred and the details surrounding the attack such as damage, cause and enemy COA identification.

73

The analysis of the data should break down the findings from collection and deduce their meaning. The analysis should compare the collected evidence information to the EML to determine if the attack actions identified in the DCBDA preparation phase occurred.

3.4.3. *Reporting*

The report is the final product of the DCBDA process, and is the detailed chronicle of the damage assessment based on the evidence analysis. The purpose of this report is to document the determination if an attack, or weapon system, was used against the target. An analyst or examiner has the responsibility for completely and accurately reporting the findings and results of an analysis of the digital evidence examination [DoJ04]. This step relies on the presence of proper documentation at each step of the DCBDA process. Notes, copies, times, documents and any pertinent additional information must be kept throughout the assessment.

The report lists the collected data from the assessment. This data will include the list of evidence found from collection and the associated attacker action which caused the evidence to be generated. Potential attack vectors, system weaknesses, root causes and incident categories should also be included in this report. The report will detail the findings of the analysis and finalize the DCBDA process.

3.4.4. *Assumptions*

This research does not address a specific organizations forensic analysis capabilities or ability to comprehend, act or report on the data collected from the DCBDA process. The actions in this methodology are used as research and meant to be tailored to

74

an operational level. For the purposes of this research it is assumed that the tools used and run during collection are done so in a forensically-sound manner.

3.5.   Experimental Method to Verify the DCBDA Process

The objective of the experiment outlined in this Section is to verify the DCBDA process by conducting assessments as outlined in the methodology in Sections 3.3 and 3.4. The experiment will create a CAMEL/CAMAT, and then conduct different DCBDAs on target test systems. The EML created during the preparation phases will then be compared to the gathered evidence to determine the success of the DCBDA. The test systems for use in this experiment will represent computer systems which are the victims of successful cyber attacks. Appendix H details the steps taken to setup the test systems. The experiment is divided into three tests each designed to test an attack scenario produced from the DCBDA preparation phase.

3.5.1. *Experiment Scope*

The attack used for this experimentation will focus on the *Covering Tracks* attack action of the cyber attacker methodology [SkL05]. This decision places a margin on the experiment, as that any attack tactic, technique, procedure and target could be used as a basis for an attack tree model and subsequent steps of the DCBDA process. Focusing on one attack allows for a threshold to be placed on the amount of data included for a viable proof of concept of the modeled attack tree, the forensic marker enumeration, analysis and subsequent actions in other areas in this research.

75

The target OS used for this attack is Windows XP. The attacker capabilities do not include any attack action or method which is evaluated to be of high risk of discovery. Also, any action which is outside the scope of the target OS is not be considered.

3.5.2. *System Boundaries*

The System Under Test (SUT) is the DCBDA process. Figure 19 shows the SUT. It consists of the AAT (TAAT), the ASL, the EML and Forensic Analysis components. The Component Under Test (CUT) is the data contained in the EML of the DCBDA. The CUT is designed and optimized for each test to compare system output to the workload parameter. The workload parameter of the system is an attack scenario environment, such as a victim computer system. The system parameters are the CAMEL and CAMAT. The system response metric is the gathered evidence, in a DCBDA report, which documents the damage and evidence of the cyber attack event.
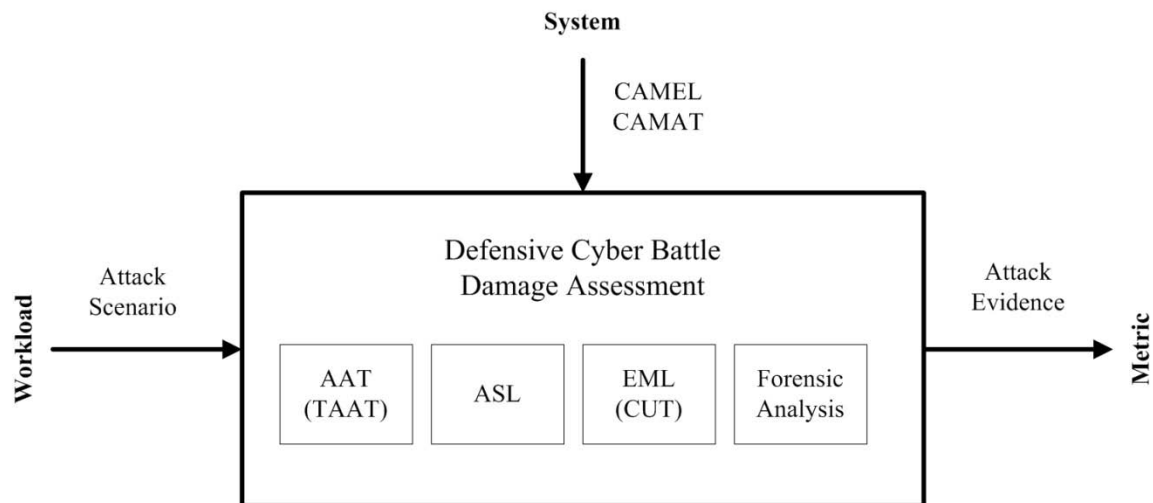
Figure 19. DCBDA SUT Diagram.

3.5.3. *Experimental Design*

The objective of the experiment is to determine if the DCBDA process is able to conduct a conclusive cyber attack damage assessment. This is done by comparing the EML of the test to the evidence gathered from conducting a DCBDA on the test victim computer.

The first step of the experiment is to create a CAMEL and CAMAT from the methodology steps outlined in Section 3.2. These products are used for all of the tests without modification.

The next step of the experiment is to conduct the preparation phase of the DCBDA process as outline in Section 3.3. This step of the experiment will use the CAMEL and CAMAT created in the previous step and the attack scenarios outlined in Table 21 as attack event input. This step of the experiment must create an ASL for each of the three attack scenarios. These three ASLs will be used to produce three respective EMLs, one for each forensic analysis experiment tests.

Table 21. Experiment Test Scenarios.

Attack Scenario	Initial Analysis Attack Action	Damage suspected
1	Covering Tracks maneuver has occurred	A suspected ICMP covert channel is disguised by a rootkit. Possible log manipulation suspected as well.
2	Covering Tracks maneuver has occurred	Possible auditing manipulation. Alternate data stream possible to cover attack executables. Files denoting recent activity may have been cleaned off of system as well.
3	Covering Tracks maneuver has occurred	Suspicious files on the target system

77

The final step of the experiment is to conduct the forensic analysis phase of the DCBDA process. The inputs for this part of the experiment are the three EMLs created in the previous step. These experiment tests will produce reports which will be compared to the EMLs to determine the results of the experiment as a whole. The forensic analysis experiment tests for this step are described below.

3.5.3.1. Test 1

This test uses the system Test_1 described in Table 22. The DCBDA input is attack scenario 1 in Table 21. The DCBDA forensic analysis phase will be completed as outlined in Section 3.4 using EML 1 produced in the DCBDA preparation phase following the previous step of the experiment.

Table 22. Experiment Test_1 System.

VM Name	Test_1
VM Configuration	2.4GHz Intel Core 2 Quad CPU using 2 cores, 1GB RAM, 3GB hard drive, bridged Ethernet adapter
Guest OS	Windows XP Service Pack 3
Specific Attacker Actions/Files	(1) WinPCap 4.1.2 installed. (2) ptunnel.exe used to create an ICMP tunnel to a receiving proxy previously setup. (3) The FU rootkit used to hide the ptunnel.exe process. (4) Evidence Eliminator used to clear the event logs of any trace of these actions.

3.5.3.2. Test 2

This test uses the system Test_2 described in Table 23. The DCBDA input is attack scenario 2 in Table 21. The DCBDA forensic analysis phase will be completed as

outlined in Section 3.4 using EML 2 produced in the DCBDA preparation phase following the previous step of the experiment.

Table 23. Experiment Test_2 System.

VM Name	Test_2
VM Configuration	2.4GHz Intel Core 2 Quad CPU using 2 cores, 1GB RAM, 3GB hard drive, bridged Ethernet adapter
Guest OS	Windows XP Service Pack 3
Specific Attacker Actions/Files	(1) Disable EventLogs service in Administrator tool: Services. (2) Two Alternate Data Streams (ADS) created using the type cmd to hide notional attack executable bad.exe in calc.exe. (3) Two txt files created, deleted and 'emptied' from Recycle Bin. (4) ZeroTracks used to remove browsing and recent file history.

3.5.3.3.   Test 3

This test uses the system Test_3 described in Table 24. The DCBDA input is attack scenario 3 in Table 21. The DCBDA forensic analysis phase will be completed as outlined in Section 3.4 using EML 3 produced in the DCBDA preparation phase following the previous step of the experiment

Table 24. Experiment Test_3 System.

VM Name	Test_3
VM Configuration	2.4GHz Intel Core 2 Quad CPU using 2 cores, 1GB RAM, 3GB hard drive, bridged Ethernet adapter
Guest OS	Windows XP Service Pack 3
Specific Attacker Actions/Files	(1) Create secrets.txt file. (2) Use steghide-0.5.1-win32 to hide secrets.txt into winter.jpg. (3) Use timestomp.exe to modify time value of winter.jpg to Monday 1/1/2001 01:01:01 AM.

79

3.5.4. *Test Systems Specifications*

Each test machine, which is a clean system before attacker actions, is a virtual machine instance of Windows XP, Service Pack 3, running on top of a 64-bit Microsoft Windows 7 Enterprise host operating system. VMWare Workstation 7.1.8 build 324285 provides the virtualization support.

Using virtual test systems as defined in Tables 22, 23 and 24 allows the attack actions to run on the clean system, and then the system state can be reverted for the next attack test. This also provides the capability to create an image of the machine's state at any time during the tests. This allows for restoration of the machine's state the clean baseline or test state for evidence collections.

3.6. Methodology Summary

In this chapter the methodology to complete the research is outlined. The approach to create the CAMEL is discussed to include how to gather and model information for the CAMAT. The DCBDA process is outlined. The basic steps to analyze the attack data and the selection of attack scenarios are discussed. The evidence evaluation process of DCBDA is covered to include the Evidence Marker List. The basics of the forensic collection, analysis and reporting are also outlined. The Section concluded by covering the experimental design.

80

IV. Experimentation and Results

The purpose of this chapter is to document the experimentation and results from applying and testing the DCBDA process as documented in Chapter 3. The focus of this chapter is to create a CAMEL and CAMAT, perform the DCBDA preparation phase actions, and conduct the DCBDA forensic analysis tests. The conducted DCBDA forensic analysis will use the EMLs from the preparation phase to actively conduct assessment experiments on target test systems to determine the validity and success of the DCBDA process. The emphasis, while conducting these actions, focuses on analyzing the results and documenting relevant findings.

Section 4.1 covers the creation of the CAMEL and CAMAT. Section 4.2 covers the completion of the DCBDA preparation phase. Section 4.3 reports the conducting and results of the forensic analysis experiments. Section 4.4 is an overall representation of the experiment results. Section 4.5 presents a summary of the chapter.

4.1. Create the CAMEL and CAMAT

The data in the CAMEL is essential to the DCBDA process. For this experiment, a CAMEL is constructed and used to prepare for the active collection during the forensic analysis experimentation. The results from following the CAMEL and CAMAT creation methodology for this experiment are outlined in the following Section.

81

4.1.1. *CAMEL Attack Action Data Collection*

The objective to this step of building the CAMEL is to collect data regarding the cyber attack methodology in its entirety. To gather information concerning this attack, CAMEL Attack Action Data Forms are filled out and the data is coalesced into a spreadsheet as shown in Appendix B. A variety of sources are used for the attack action information gathering [Bos02] [DBM09] [MSK05] [SkL05]. The attack vector information included uses Appendix A as the format for the data. A subset of the gathered data is shown in Table 25.

Table 25. CAMEL Attack Action Data Subset.

Attack Action	Parent Action(s)	Child Action(s)	Attack Vector	Attack Results/Goal
ICMP Covert Channel	Covert Channel	None	4a,4b	This action installs an ICMP covert channel
HTTP Covert Channel	Covert Channel	None	4a,4b	This action installs a HTTP covert channel
DNS Covert Channel	Covert Channel	None	4a,4b	This action installs a DNS covert channel
TCP Covert Tunnel	Covert Channel	None	4a,4b	This action installs a TCP covert channel

A threshold is placed on the amount of data collected for the CAMEL used in this experiment. The data gathered concerning the cyber attack methodology represents a possible best effort comprehensive collection with the attention for greater detail placed on the *Covering Tracks* attack actions. The data collected facilitates the needs for the remaining steps of this experiment.

4.1.2. *CAMEL Attack Method Analysis*

With the attack actions input into CAMEL, the actions are now analyzed to link and associate appropriate attack methods capable of creating the actions. For this experiment, this data is discovered through a variety of sources which are included in the spreadsheet located in Appendix C. This spreadsheet contains the results of the CAMEL attack method analysis and extends the data listed in the previous step. The data is considered to be a representation of a possible best effort analysis of the attack actions. Particular effort is placed on analyzing the *Covering Tracks* attack actions. Table 26 shows a subset of the attack method data.

Table 26. CAMEL Attack Method Data Subset.

Attack Method	Attack Action	Attack Method Description	Source
cmd: cp	Alternate Data Stream	Command line type	http://www.windowsecurity.com/articles/Alternate_Data_Streams.html
AFX Windows RootKit	Application Rootkit	System patch to hide information	http://www.megasecurity.org/trojans/a/aphex/Afx_win_rootkit2003.html
Evidence Eliminator	Delete Attack Files	Erase temp files, histories, recent documents	http://www.evidence-eliminator.com/
Tracks Eraser Pro	Delete Attack Files	Erase temp files, histories, recent documents	http://www.acesoft.net/
Netcat	Backdoor	Backdoor remote access program	http://www.securityfocus.com/tools/139
ZeroTracks	Delete browsing history	Erase temp files, histories, recent documents	http://zerotracks.en.softonic.com/

4.1.3. *CAMEL Evidence Analysis*

The CAMEL evidence analysis process conducts an analysis on the attack methods and actions to identify the forensic evidence markers which act as a digital

83

fingerprint of the attack. For this experiment, this data is discovered through a variety of sources which are included in the spreadsheet located in Appendix D. The evidence confidence evaluation values are decided through consideration of the evidence source and are based on a cursory analysis of the markers ability to identify the associated attack method or action. The metrics used are low, medium or high confidence factor. Table 27 shows a subset of the data in Appendix D.

Table 27. CAMEL Evidence Analysis Data Subset.

Evidence Name	Attack Action/Method	Forensic Marker	Forensic Tools	Evidence Confidence	Evidence Source
Failed login attempt	Brutus	Failed login attempts in log	PyFlag	H	http://www.webhostgear.com/240.html
Proxy attack	WebScarab	Malformed data	Wireshark	H	http://www.owasp.org/index.php/Category:OWASP_WebScarab_Project
suspicious open ports	Netcat	suspicious open ports	netstat	H	http://technet.microsoft.com/en-us/library/bb490947.aspx
Trojan actitivy	Beast	Suspicious Trojan activity	Regedit.exe	H	http://www.exterminate-it.com/malpedia/remove-beast#howfiles
System folder missing	cmd: del windows\prefetch	windows\prefetch missing	cmd	H	None
Hidden folders	Hide Folders 2009	Hidden folders	cmd	L	http://www.fspro.net/hide-folders/

The spreadsheet in Appendix D contains the results of the CAMEL evidence analysis. This product extends the CAMEL created in the previous steps. The data values assigned for the evidence analysis portion of the experiment are considered to be a representation of a possible best effort analysis of the attack methods. Particular effort is placed on analyzing the *Covering Tracks* attack methods.

4.1.4. *CAMEL Forensic Tool and Method Analysis*

This step of the CAMEL creation process identifies the forensic method or tool which has the ability to retrieve the identified forensic evidence markers. Each unique forensic marker is analyzed for a forensic retrieval method. These identified tools are essential to the forensic analysis process and due consideration should be given to analyzing the correct tool to retrieve the attack evidence.

The spreadsheet in Appendix E contains the results of the CAMEL tool and method analysis for this experiment. This product further extends the CAMEL created in the previous steps. The data values assigned for the tool and method analysis portion of the experiment are considered to be a representation of a possible best effort analysis and evaluation. Particular effort is placed on analyzing the *Covering Tracks* evidence markers. Table 28 shows a subset of the data captured for the method and tool analysis.

Table 28. CAMEL Forensic Tool and Method Data Subset.

Forensic Marker	Forensic Tool/Method	Tool Confidence	Tool/Method Source
Streams will examine the files and directories for streams	Streams	M	http://technet.microsoft.com/en-us/sysinternals/bb897440.aspx
Searches drives & lists all files that have ADS	ADSTools	M	http://www.soft32.com/download_207535.html
System search for all active ADS	ADS Scanner	M	http://www.pointstone.com/products/ADS-Scanner/
list, view or delete Alternate Data Streams (ADS)	ADS Spy	M	http://www.brothersoft.com/ads-spy-74079.html
Suspected Rootkit activity	RootkitRevealer	H	http://technet.microsoft.com/en-us/sysinternals/bb897445

85

4.1.5. *CAMEL Attack Action Metrics Analysis*

This step of the CAMEL creation process analyzes the combined aggregation of the data in the CAMEL to assign capability metrics to each attack action. These metrics must be based off of all data associated to the action. This is namely the attack method, evidence and tool/method for retrieval.

For this experiment two metrics are assigned to each attack action. Risk of attack discovery is a low, medium or high value given to the action which is a value given for the likelihood the attack action is discovered in a typical target computer system. The values assigned are based on a computer system being passively monitored for unauthorized activities by some means. The second value is an impact of attack measured as low, medium or high value given to the action which is a value of what impact the action has on a target system. Each of the attack actions are analyzed and assigned a value for these factors. Table 29 contains a subset of the data collected for this analysis.

Table 29. CAMEL Capability Metrics Data Subset.

Attack Action	Parent Action(s)	Child Action(s)	Risk of Discovery	Impact
ICMP Covert Channel	Covert Channel	None	M	H
HTTP Covert Channel	Covert Channel	None	H	H
DNS Covert Channel	Covert Channel	None	L	H
TCP Covert Tunnel	Covert Channel	None	H	L

The spreadsheet in Appendix F contains the results of the CAMEL attack action metrics analysis. This product extends the CAMEL created in the previous steps. The data

values assigned for the attack action metrics analysis portion of the experiment are considered to be a representation of a possible best effort analysis and evaluation. Particular effort is placed on analyzing the *Covering Tracks* metrics.

4.1.6. *CAMAT Model*

The experimental data collection and analysis for CAMEL is complete. Now the data is modeled as an attack tree. The root goal used for CAMAT is *Cyber Attack*. The tree building process outlined in Section 3.2.6 is followed to build the CAMAT. The completed model is shown in Figure 20, with the majority of the *Covering Tracks* portion of the attack tree not shown. That attack area will be covered in greater detail during the DCBDA process of the experiment.
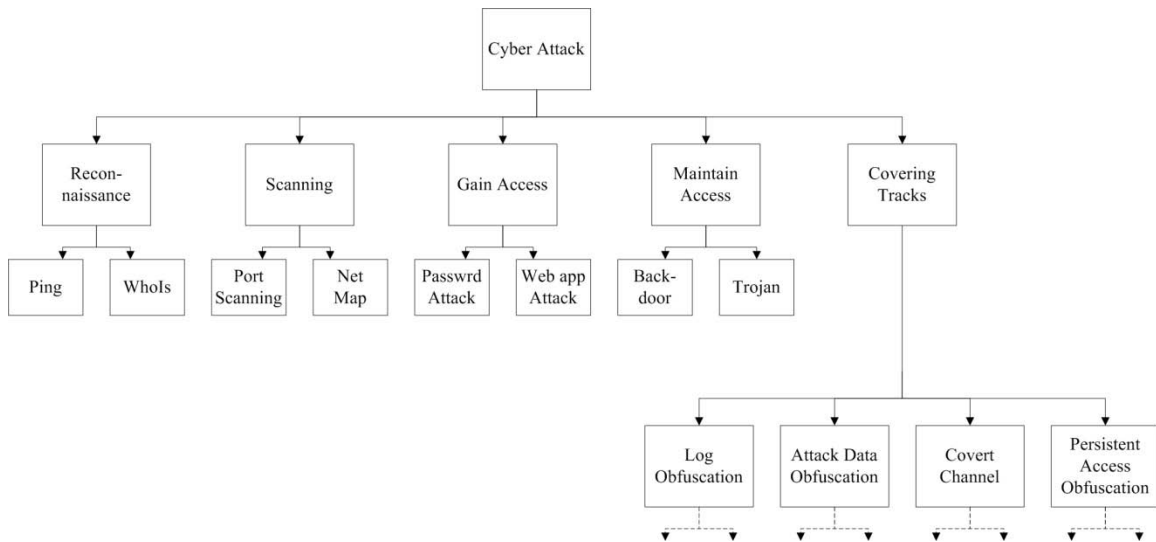
Figure 20. An Example of a CAMAT.

4.1.7. *CAMEL Findings and Results*

The research and creation of the CAMEL and CAMAT following the methodology outlined in Chapter 3 led to important findings. These findings can be used to further refine this research, be considered for operational CAMEL usage, or help guide future work efforts. The overall result is the creation of the CAMEL and CAMAT for use in the DCBDA experimentation.

4.1.7.1.   *Attack Action Data*

Overall, the information included in CAMEL must be tailored to fit an organization's operational requirements for DCBDA. The adaptation of the CAMEL process in this research must be thoroughly managed and documented starting with the CAMEL Attack Action Data Form. This form is the basis for all other CAMEL data collection and analysis. The data and format for this initial population of information into the CAMEL must be given due consideration in an operational implementation to ensure the success of dependent cyber battle damage assessments.

4.1.7.2.   *Evidence Marker Analysis*

During this research implementation of CAMEL it is important to ensure evidence markers are as detailed and accurate as possible. These markers are the pivotal data of the CAMEL. The forensic tool must be properly identified from the evidence marker analysis and also have the ability to properly find the correct data to assess damage from a cyber attack. This can only happen through the detailed analysis of attack evidence markers.

88

4.1.7.3.   *Sensitive Data*

The detailed collection and analysis of an adversaries attack methodology actions should be considered sensitive information.   An operational implementation of the CAMEL and CAMAT should consider this sensitivity of the data aggregation.   This product could also have the potential to be used against an organization if the attacker gained knowledge of their understanding of the cyber attack.

4.2.   DCBDA Preparation

In this Section the preparatory phase actions of the DCBDA is completed. This is done   by   collecting   attack   event   information,   building   the   AAT   from   the CAMEL/CAMAT,   identifying   likely   attack   COAs   through   capability   analysis,   and evaluating the relevant evidence to create the EML for forensic analysis consideration.

4.2.1. *AAT Creation*

The root goal attack being used for this experiment is the *Covering Tracks* phase of the hacker methodology which seeks to obfuscate any actions taken during the rest of the attack [SkL05]. This is the root goal for the AAT used for the DCBDA in this experiment. To build the AAT the *Covering Tracks* attack action is found in the CAMEL and the CAMAT.   All data concerning the *Covering Tracks* attack action and the subordinate children are taken from the CAMAT and used to build the AAT.

The AAT of the *Covering Tracks* attack action is shown in Figure 21. The AAT has 26 leaf nodes identified and 4 main children; log obfuscation, persistent access obfuscation, attack data obfuscation and covert channel. Now that the AAT is created

89

using the data in CAMAT and CAMEL, the attack methodology capability analysis of the attack tree metrics is applied.

4.2.2. *AAT Analysis and TAAT Creation*

The key step of this action in the DCBDA process is to apply an analysis to the data which has been retrieved from the CAMEL and modeled in the AAT. This is done through capability determination and analysis. For this experiment, the metrics in the AAT built from the CAMEL are not altered as part of the capability determination process.

The capability analysis used for the TAAT follows the experimental scope of the attacker capabilities. For this experiment, any action which had a *high* risk of discovery is removed from the attack tree. Once the metric analysis is complete a pruning, or removal or irrelevant actions, of the attack tree is accomplished.

The pruned AAT, now called the TAAT, is then modeled with only the nodes deemed the focus of this DCBDA. The *Covering Tracks* TAAT model is shown in Figure 22. The TAAT, along with the associated data for each action from CAMEL, will be the input for the next part of the experiment, the DCBDA COA analysis.
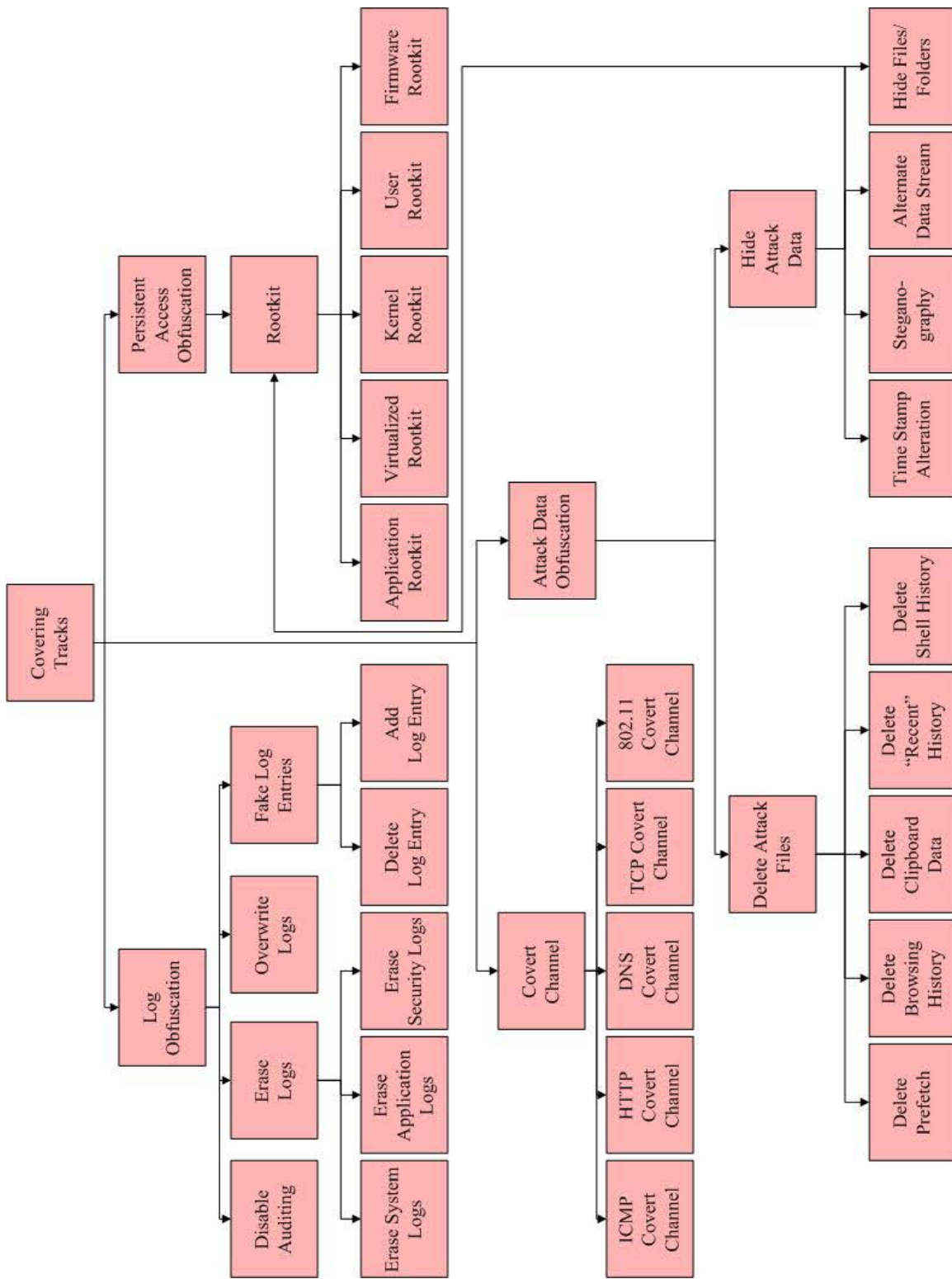
Figure 21. Covering Tracks AAT.
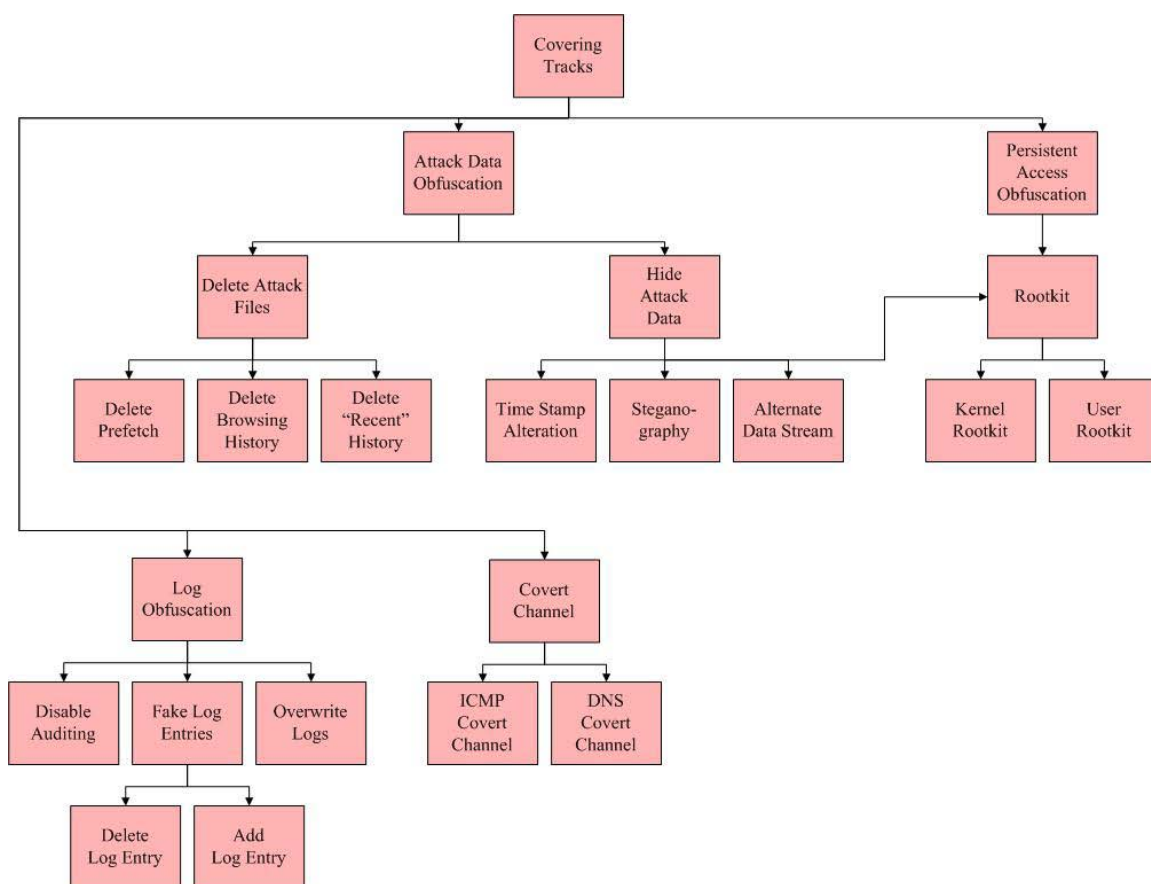
Figure 22. Covering Tracks TAAT.

4.2.3. *COA Analysis and ASL Creation*

Using the knowledge of the attack from the AAT analysis, the goal of this step is to understand and produce tailored attack COA intelligence, namely the listing of prioritized attack scenarios tailored to the attack being assessed. This list is captured in the ASL. The attack scenarios are created as a listing of attack actions an attacker would use during an attack.

Table 30 shows the ASL which has the three attack scenarios created for this experiment. These attack scenarios represent three plausible situations for this

92

experiment following the experimental scope. The shaded cells represent leaf nodes. Attack Scenario 1 is a covert channel being the first step taken by the attacker. Attack Scenario 2 focuses on auditing disablement as a priority. Finally, attack scenario 3 focuses on steganography use.

Table 30. DCBDA Experiment ASL.

Scenario 1	Scenario 2	Scenario 3
Covering Tracks	Covering Tracks	Covering Tracks
Covert Channel	Log Obfuscation	Hide Attack Data
ICMP Covert Channel	Disable Auditing	Steganography
Persistent Access Obfuscation	Hide Attack Data	Hide Attack Data
Rootkit	Alternate Data Stream	Time Stamp Alteration
Kernel Rootkit	Delete Attack Files	
Log Obfuscation	Delete Prefetch	
Erase Logs	Delete Browsing History	
	Delete "Recent" History	

4.2.4. *Evidence Evaluation and EML Creation*

The goal of evidence evaluation in the DCBDA process is to use the listed attack scenarios in the ASL to identify the forensic evidence markers and the forensic tools from the CAMEL necessary to determine if the actions took place on a system. This data is input into the EML which is the product of the evaluation and used by the DCBDA forensic analysis to determine if the attack scenario occurred. The three EMLs created from this data are shown in Tables 31, 32 and 33.

93

Table 31. EML for ASL: Scenario 1.

Evidence Name	Attack Action/Method	Forensic Marker	Forensic Tool
ptunnel	ptunnel	ptunnel.exe on system	cmd, search
ICMP traffic volume	ICMP Covert Channel	Suspicious ICMP traffic	Wireshark
WinPCap for ptunnel	ptunnel	WinPCap installed	cmd, search
Kernel Rootkit	Rootkit	Suspected Rootkit Activity	RootkitRevealer, FSecure Blacklight
Missing/Incomplete logs	ClearLogs	Missing/incomplete logs	Event Viewer

Table 32. EML for ASL Scenario 2.

Evidence Name	Attack Action/Method	Forensic Marker	Forensic Tool
Missing/incomplete logs	Disable Auditing	Missing/incomplete logs	Event Viewer
Disable EventLog service	Disable Auditing	Disable EventLog service	Services
ADS detection	Alternate Data Stream	System search for all active ADS	ADS Scanner
Files removed	ZeroTracks	Files removed	File Scavenger

Table 33. EML for ASL Scenario 3.

Evidence Name	Attack Action/Method	Forensic Marker	Forensic Tool
Suspicious files/sizes	Steganography	Suspicious files/sizes	Stegdetect
Suspicious files/sizes	Steganography	Suspicious files/sizes	StegSpy
Suspicious time stamp values	Timestomp	Suspicious MACE values	dir c:\ /A /S /T:W

4.2.5. *DCBDA Preparation Findings and Results*

The results of this stage of the experiment are the three EMLs which contain the evidence markers and tools necessary to identify the attack under assessment in the experiment tests. The results achieved from executing the cyber attack methodology in Chapter 3 match the expectations of the thesis, which is the completion of the preparation phase of the DCBDA. The process relies heavily on the work completed for the CAMEL. Notable findings during the conduct of the experiment are listed here.

94

4.2.5.1.   Capability analysis is crucial to proper assessment

The metrics given to an attack node should be chosen and assigned with proper consideration.   These metrics are the basis for tree pruning and subsequent attack scenario selection.   Improper values for these metrics can lead to incorrect pruning of possible attack actions and therefore the damage assessment will be hindered.

4.2.5.2.   EML data

The data in the EML should be as relevant as needed for the organization performing the forensic analysis.   This EML format should be modified to include any additional information needed from the DCBDA preparation phase.

4.3.   Experiment to Verify the DCBDA Process

This portion of the experiment conducts the DCBDA forensic analysis process on the test systems described in Section 3.5.3 using the methodology outlined in Section 3.4 The goal of this portion of the experiment is to test the CUT, the three EMLs created in Section 4.2.4, to the results of the these tests. The objective is to determine if a *Covering Tracks* maneuver took place on the target test systems and to also deduce the specific attacker actions discovered.

4.3.1. *Test 1 Collection and Analysis*

The tools identified in the EML for Scenario 1 in Table 31 are run on the system Test_1 as described in Table 22.   The results are listed below.

95

The first method for action is searching for the ICMP covert channel program, ptunnel. The resulting search found existence of files associated with the tool. The results are shown in Figure 23. The existence of ptunnel in the \Prefetch folder indicates that the program has been executed on Test_1.

Figure 23. Search Results for ptunnel on Test_1.

The second method in the EML is to use the tool Wireshark tool to identify suspicious ICMP traffic. The tool is installed and used on Test_1. The results in Figure 24 show the tool is able to identify suspicious ICMP traffic. The ICMP traffic to an unknown address indicates Test_1 is using a covert channel to communicate undetected.

Figure 24. Results for Wireshark Execution on Test_1.

96

The third method identified in the EML is to search for WinPCap installed on the target. The basic search found existence of files associated with the WinPCap program. The results are shown in Figure 25. This is another indication that a covert channel exists.
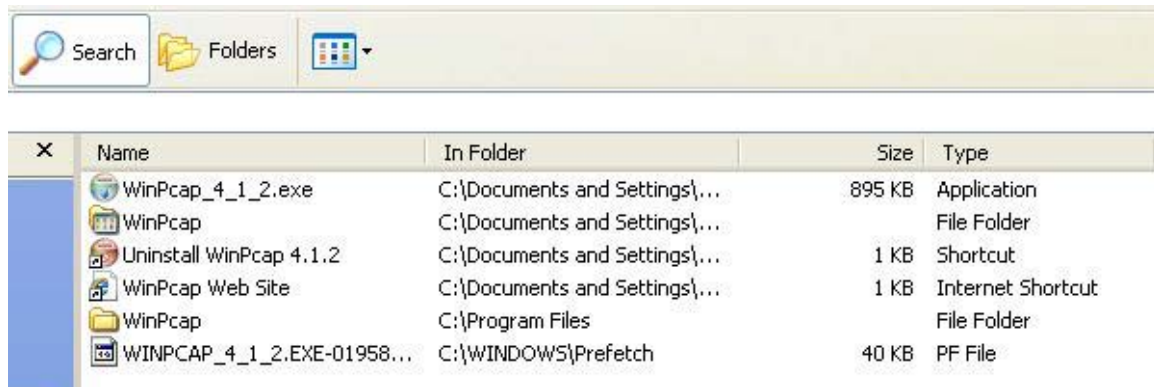
Figure 25. Search Results for WinPCap on Test_1.

The fourth method in the EML for test 1 is to use RootkitRevealer and F-Secure Blacklight to search for a suspected Rootkit. F-Secure Blacklight found the ptunnel.exe process which is hidden. The process identified by Blacklight is shown in Figure 26.

Figure 27 shows the results from running RootkitRevealer on the target system. RootkitRevealer is able to identify two suspect registry keys. Overall the results for this step in the EML are a success, due to Backlight's identification of the hidden covert tunnel process.

The final method in the EML is to use Event Viewer to discover any missing or incomplete logs. Figure 28 shows the success of finding the Security log empty. This log being modified is a positive indication of *Covering Tracks* evidence.

97
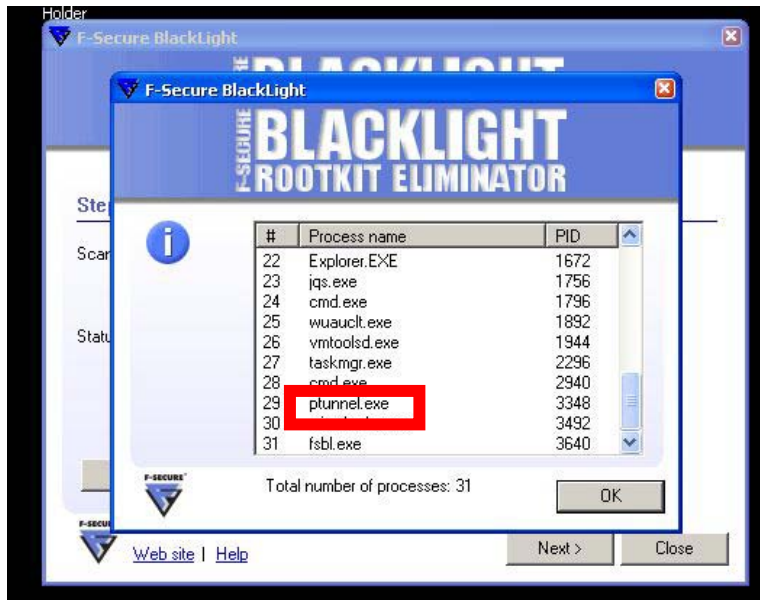
Figure 26. F-Secure BlackLight Results for Rootkit Detection on Test_1.
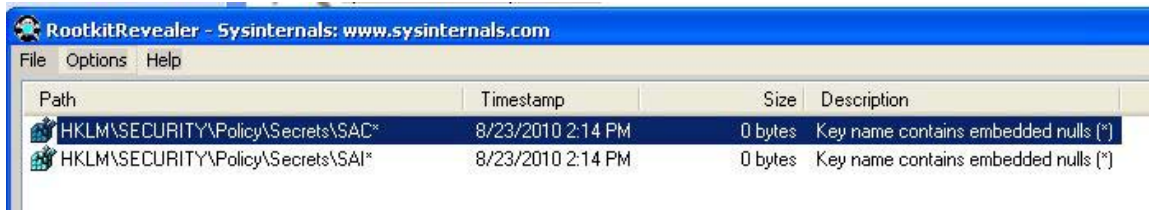
Figure 27. RootkitRevealer Results on Test_1.
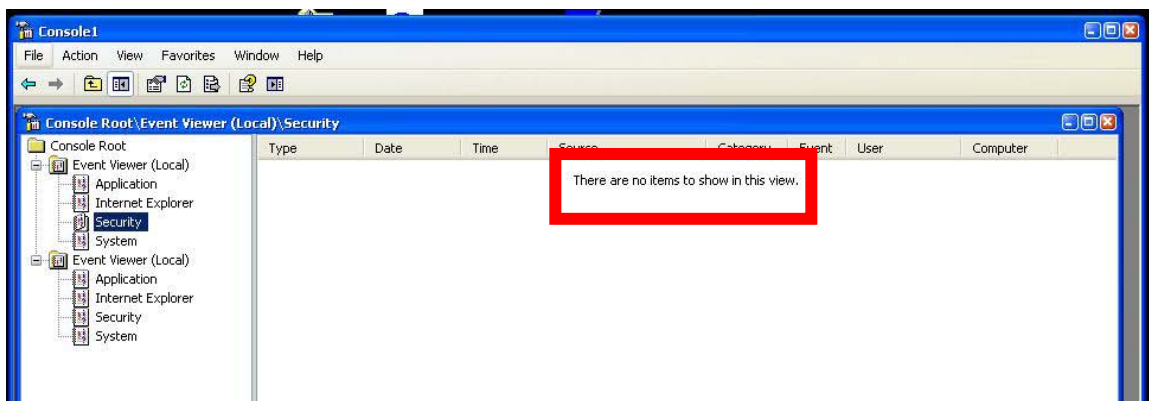
Figure 28. Event Viewer Results on Test_1.

98

4.3.2. *Test 1 Report and Results*

This Section is the forensic analysis report for the first test and is the final product of this DCBDA process. The determination of the assessment is a successful cyber attack is executed against the target. The ptunnel and FU Rootkit weapons were used in a *Covering Tracks* maneuver.

Test 1 is a successful experiment of the ability of the DCBDA process to create a tailored EML for the attack scenario. This EML is used to collect and analyze specific evidence found on the target system. The evidence results, summarized in Table 34, show the attack scenario of a successful ICMP covert channel hidden by a rootkit and obfuscated by log manipulation. This resulted in the positive identification of all forensic markers indicating a successful cyber attack action of *Covering Tracks*.

Table 34. Test 1 Results.

Forensic Marker From EML	Collection/Analysis Success
ptunnel.exe on system	Yes
Suspicious ICMP traffic	Yes
WinPCap installed	Yes
Suspected Rootkit Activity	Yes
Missing/incomplete logs	Yes

4.3.3. *Test 2 Collection and Analysis*

The tools identified in the EML in Table 32 are run on the system Test_2 as described in Table 23. The results are listed in this Section.

99

The first method for action is to use Event Viewer to identify any missing or incomplete logs. Figure 29 shows the results of using Event Viewer to inspect the event logs. The error is a clear indication of a log obfuscation attack action.
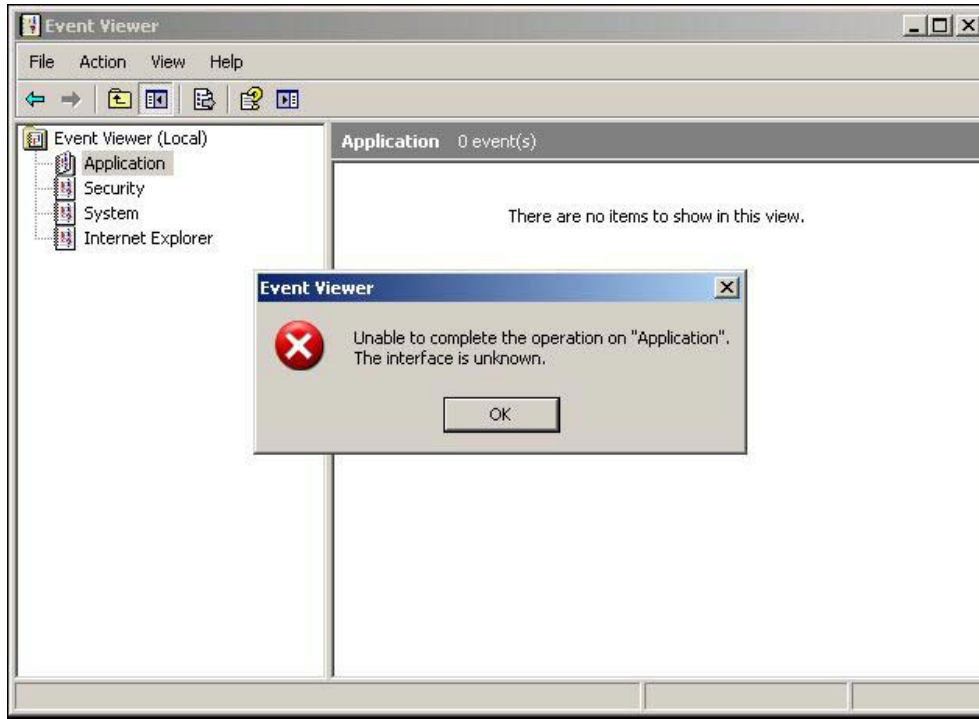
Figure 29. Event Viewer Results on Test_2.

The second method to use to collect evidence from Test 2 is the administrator program Services. The EventLog service is inspected and Figure 30 shows the results. The service being disabled is a indicator of a log obfuscation attack action.

100

Figure 30. Services Results on Test_2.

The third method for collection actions is to use ADS Scanner to search for all active ADS. The scanner is loaded and run on Test 2. The findings in Figure 31 depict two active ADS on the system. This activity is suspicious and the name of *bad.exe* is an indication of a cyber attack.
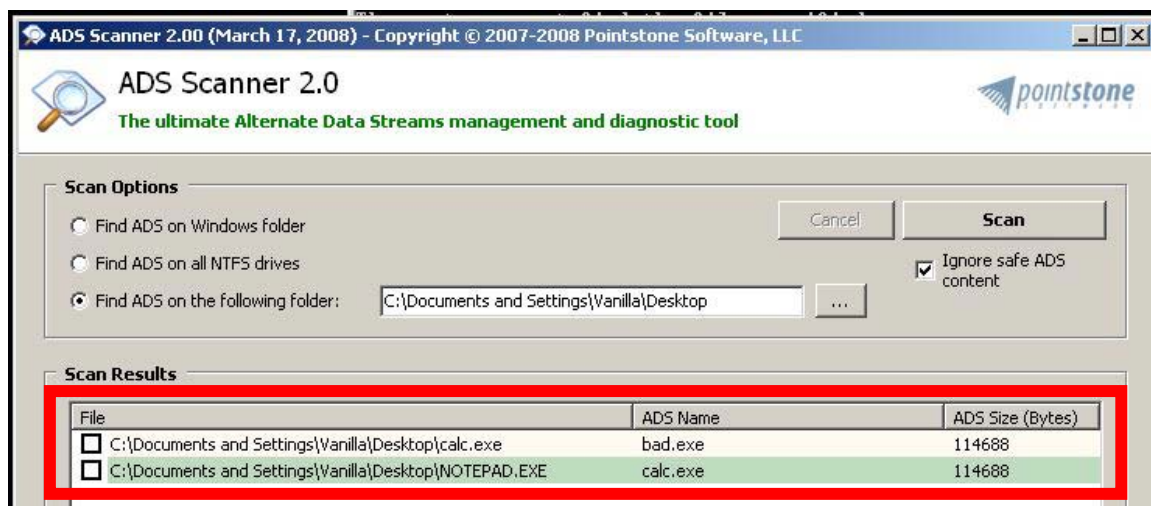
Figure 31. ADS Scanner Results on Test_2.

101

The last forensic method and tool identified for action in the EML is to use File Scavenger to identify removed files indicating an attacker presence. The tool is run on the target system and the tool identified suspicious files and recently deleted files. These actions, shown in Figure 32, show that *Company Sales Plan.lnk* and *Company Secrets.lnk* shortcuts exist, but the files they are linked to do not exist. This is an indication of an attacker attempting to delete these files as part of an attack file obfuscation action. Also discovered, as shown in Figure 33, were three documents removed from the Recycle Bin. These files were not able to be recovered, but the time stamps are very close to the previously identified links. All of the identifiers found by the File Scavenger program indicate a *Covering Tracks* cyber attack.
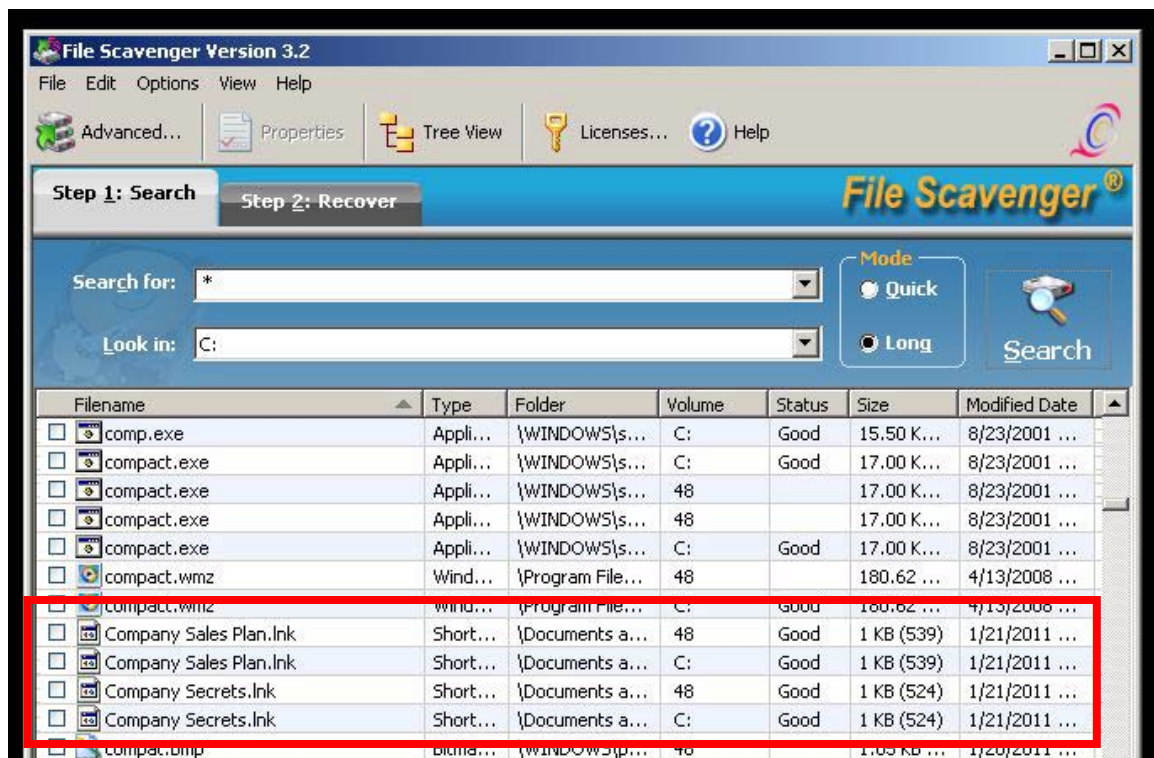
Figure 32. File Scavenger Suspicious File Links Found on Test_2.

Figure 33. File Scavenger Suspicious Deleted Files Found on Test_2.

4.3.4. *Test 2 Report and Results*

This Section is the forensic analysis report for this second test and is the final product of this DCBDA process. The determination of the assessment is a successful cyber attack was executed against the target. The EventLog service is disabled, two ADS files are discovered and suspicious documents are deleted from the system. These actions acted as the weapons used in a *Covering Tracks* maneuver.

Test 2 is a successful experiment of the ability of the DCBDA process to create a tailored EML for the attack scenario. This EML is used to collect and analyze specific evidence found on the target system. The evidence results, summarized in Table 35, show the attack scenario of an attacker covering attack action tracks by deleting files, creating ADS to hide executables and the disablement of the Event Logs service. This resulted in the positive identification of all forensic markers indicating a successful cyber attack action of *Covering Tracks*.

Table 35. Test 2 Results.

Forensic Marker From EML	Collection/Analysis Success
Missing/incomplete logs	Yes
Disable EventLog service	Yes
System search for all active ADS	Yes
Files removed	Yes

4.3.5. *Test 3 Collection and Analysis*

The tools identified in the EML in Table 33 are run on the system Test_3 as described in Table 24. The results are listed in this Section.

The first method for action is to use the program StegDetect to find any suspicious files for Steganography. The suspicious file *Winter.jpg* is identified on the user's Desktop from a basic search looking for suspicious picture files. StegDetect is used against the file with no success as shown in Figure 34. This is not an event of concern, due to the EML having more than one method identified for Steganography detection.

Figure 34. StegDetect Negative Results for Test 3.

The second method identified for use to collect evidence on Test 3 is StegSpy. This program is used on the suspicious file *Winter.jpg*. As shown in Figure 35, this identified forensic tool is not able to identify the file as having Steganography used to cover an attacker's actions.

104
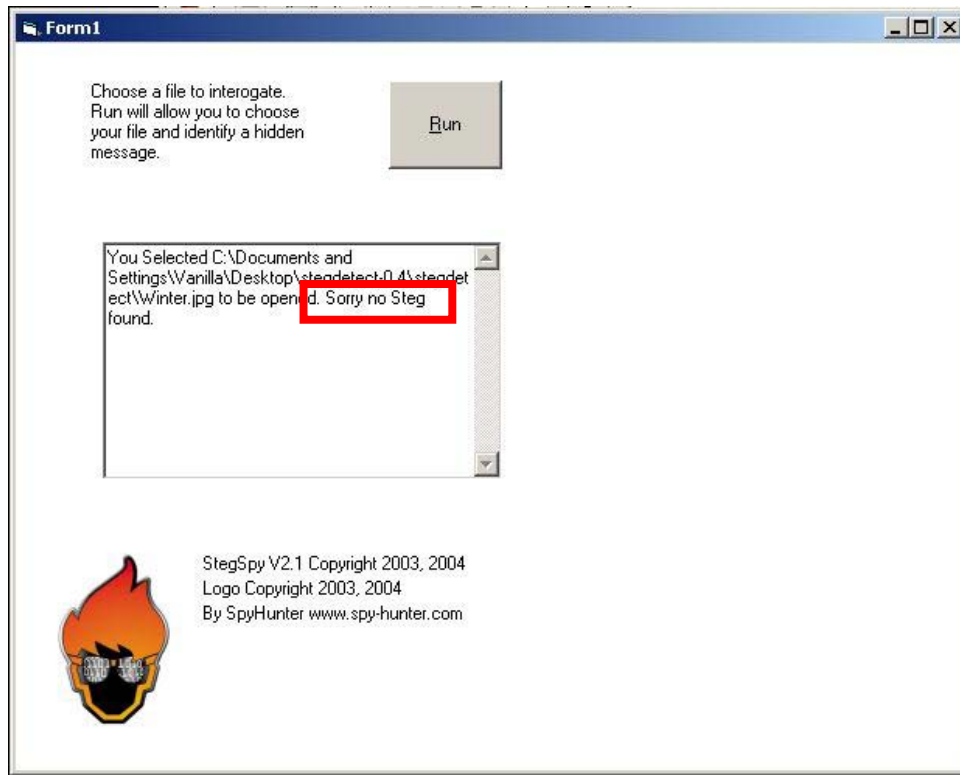
Figure 35. StegSpy Negative Results for Test 3.

The final method to identify forensic evidence is to identify suspicious time attributes associated to any file on the Test 3 system. The command "dir c:\ /A /S /T:W" is run and each time value is analyzed. The result, as shown in Figure 36, depicts the *Winter.jpg* file having a modified time value as the time is earlier than any other on the system. This indicates an attacker having modified the values to cover an attack action.
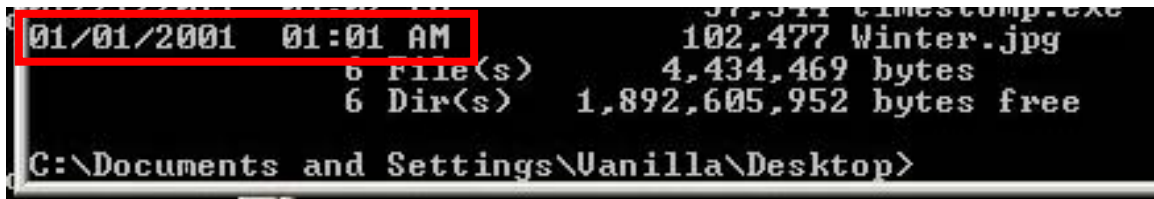
Figure 36. File Time Attribute Analysis Results for Test 3.

4.3.6. *Test 3 Report and Results*

This Section is the forensic analysis report for test number three and is the final product of this DCBDA process. The determination of the assessment is a successful cyber attack was executed against the target, with exception. The *Winter.jpg* is identified as being a suspicious file with confirmed invalid time attributes. The exception lies in the inability of the identified forensic steganalysis programs to identify a file being used to hide data. The EML still identified a *Covering Tracks* attacker action, but the finding on this test is the evidence analysis used for the EML must be accomplished again to properly identify steganography files such as *Winter.jpg*.

Test 3 is a successful experiment of the ability of the DCBDA process to create a tailored EML for the attack scenario. This EML is used to collect and analyze specific evidence found on the target system. The evidence results, summarized in Table 36, show the attack scenario of an attacker covering attack action tracks by modification of suspicious file time attributes. More work is needed to determine the extent of the forensic evidence of the cyber attack which occurred for this test.

Table 36. Test 3 Results.

Forensic Marker From EML	Collection/Analysis Success
Suspicious files/sizes	No
Suspicious files/sizes	No
Suspicious file time values	Yes

4.4. Overall Experiment Results

This Section offers an overall summary to the results of the experiment used for this experiment.

4.4.1. *Create the CAMEL and CAMAT Results: Success*

The CAMEL and CAMAT are created for the cyber attack methodology and the data is used for the experimentation of the DCBDA process. The notable findings are the importance of attack action data, the importance of proper evidence marker analysis and the sensitivity of conglomerated information concerning attack methodology.

4.4.2. *DCBDA Preparation Results: Success*

The preparation phase of the DCBDA process is conducted using the CAMEL and CAMAT. The process created the ASL with three attack scenarios created for this experiment. Three respective EMLs were created as well. The notable findings for this part of the experimentation are the importance of capability analysis and data within the EMLs.

4.4.3. *Experimentation to Verify DCBDA Results: Success With Exception*

The forensic analysis of three test systems is accomplished to complete the DCBDA process and verify if the process has the ability to properly identify a cyber attack. The experiments are a success with one exception of an inadequate EML to fully identify all factors of the evidence of the cyber attack.

107

4.4.4. *Overall Experiment Results Summary*

The results of this experimentation are summarized in Table 37.

Table 37. DCBDA Process Experimentation Results.

DCBDA EML	Forensic Evidence Match Result
Test 1	Success
Test 2	Success
Test 3	Success (with exception)

4.5. Experimentation and Results Summary

In this chapter the experiment and results are discussed. The scope of the experiment is outlined. The creation of the CAMEL through attack action collection, attack method analysis, evidence analysis, forensic tool and method analysis is covered. The modeling of the CAMAT for this experiment is shown. The completion of the DCBDA preparation phase to include attack tree creation is discussed. Finally the experimentation of the DCBDA process is conducted with three tests of the forensic analysis phase.

V. Conclusions

This chapter discusses the conclusions of the research effort and possible considerations for future research. Section 5.1 contains the research summary and objective overview. Section 5.2 provides several recommendations for extending the DCBDA research. Section 5.3 contains concluding remarks to include the contributions of this research.

5.1. Research Summary

The primary goal of this research is to provide robust defensive cyber battle damage assessment capabilities through digital forensic analysis tailored by an understanding of the attack methodology. The CAMEL is developed to facilitate this comprehensive understanding of the cyber attack. The DCBDA process is outlined and the process is verified through active forensic analysis experiments. The following Sections discuss each research objective to determine if they have been met from this research effort.

5.1.1. *Develop the CAMEL and CAMAT*

The exhaustive listing of the cyber attack methodology allows for the intelligence preparation of the operating environment to directly impact the capabilities of the DCBDA process. CAMEL gives an organization a wealth of detailed attack information to bolster their abilities to provide timely, accurate and detailed assessments of a cyber attack. This research shows the proposed CAMEL process can be used to tailor the cyber

109

damage assessment process when properly populated with accurate information. This process can increase accuracy, timeliness and capabilities of future technical damage assessments from cyber attacks.

5.1.2. *Develop the DCBDA Process*

The objective in defining the Defensive Cyber Battle Damage Assessment process is to develop a procedure which utilized a comprehensive wealth of information regarding cyber attack mythologies. The research shows the proposed two phase process allows tailored damage assessments to utilize the wealth of information in the CAMEL.

5.1.3. *Verify the DCBDA Process*

The purpose of this objective is to verify the proposed CAMEL and DCBDA processes would correctly identify a cyber attack. Through the execution of three experimental tests the processes are proven effective and able to deliver tailored damage assessment for a given cyber attack. The findings of these experiments should be used to improve the research and be considered for any operational implementation of the proposed procedures.

5.2. Future Work

Utilizing attack methodology modeling for enhanced cyber BDA has many avenues for further research. The following Section discusses possible areas for consideration to further develop this research topic.

110

5.2.1. *Real Time DCBDA*

Through the course of this research it was discovered that extending the cyber attack tree model to include more information would allow for a bottom-up identification of an attack. Future work could explore the possibility of identifying and modeling sensor alerts from a cyber attack through the CAMAT. Real time sensor alerts could then be followed up the tree to the attack tool the sensor alerted on, and then to the attack action(s) the tool is used for. From here, incident responders could use the detailed knowledge modeled in the cyber attack tree to increase their ability to pinpoint an ongoing attack. Figure 37 shows how the DCBDA process could potentially be extended to allow for this future work. This figure shows how the sensor data and attack action correlation can be found in the cyber attack tree.

5.2.2. *Mission Impact Framework Incorporation*

While it is understood that mission impact is not the same as damage assessment, mission impact analysis is still very dependent on efforts like this research. Future work could refine or further develop the DCBDA process to provide all the necessary information for the best possible integration into a mission impact framework. Research conducted by Captain Lisa S. Thiem showed in a case study of damage assessment on Air Force networks that the focus on damage assessment was exclusively on technical assessments and in some cases had no connection to higher level applications of the damage assessment [Thi05]. The DCBDA process could also be incorporated into higher level mission impact framework research [For07]. Mission impact fusion into the DCBDA process can work to alleviate this problem.

111
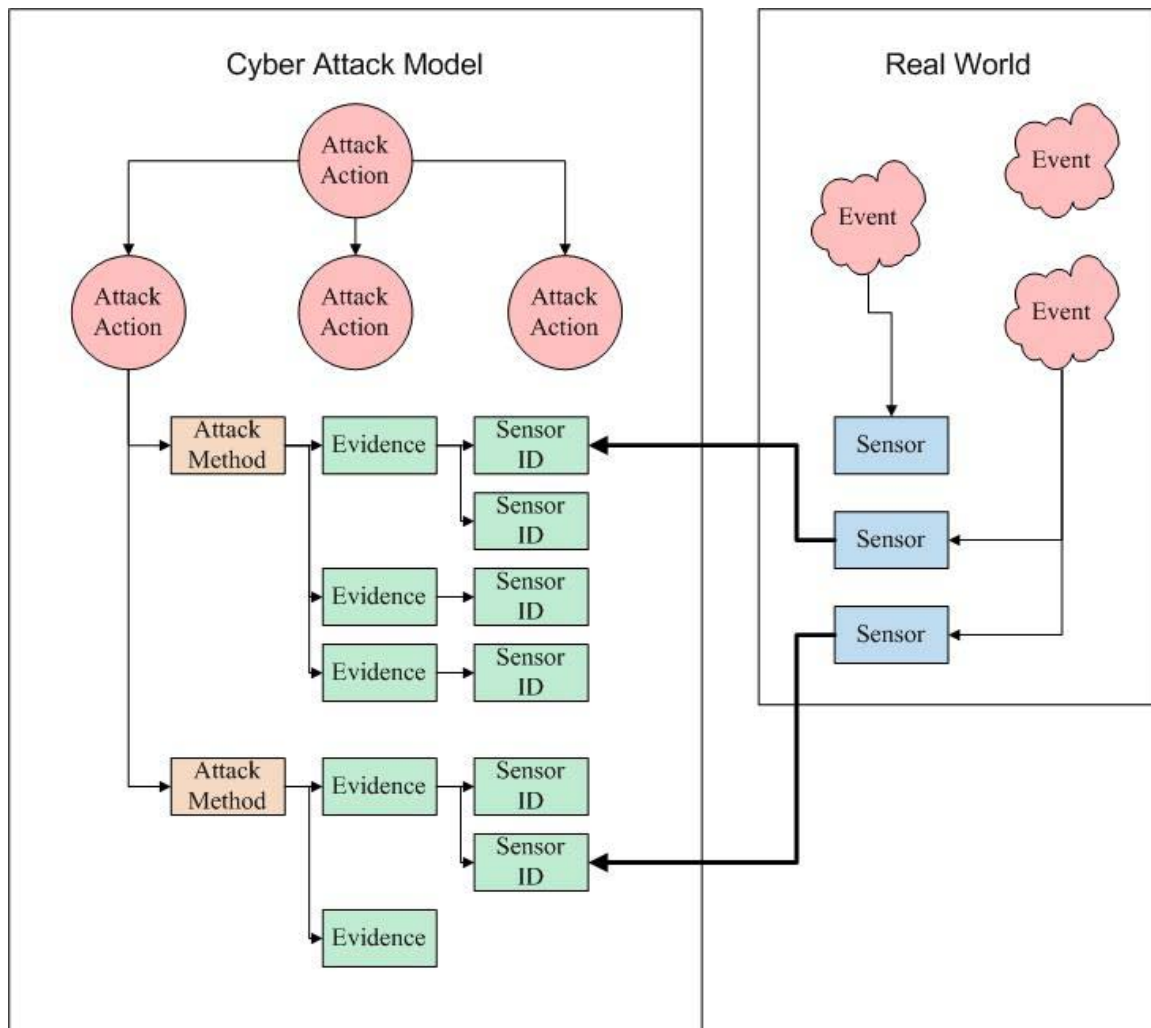
Figure 37. Real Time DCBDA.

5.2.3. *Host-Based Security System Integration*

Host-Based Security System (HBSS) is the Defense Information Systems Agency's (DISA) security framework. The ability to incorporate a battle damage assessment function into the HBSS suite which could be administered from a remote location would be extremely beneficial for incident response actions. Another system for

possible DCBDA process integration is CyberCraft [KaP07]. This integration could lead to a true attack awareness module benefiting from the DCBDA process outlined in this research. This function could be implemented as a forensic agent.

Figure 38 depicts a plausible solution to this problem building on the proposed real time DCBDA research.
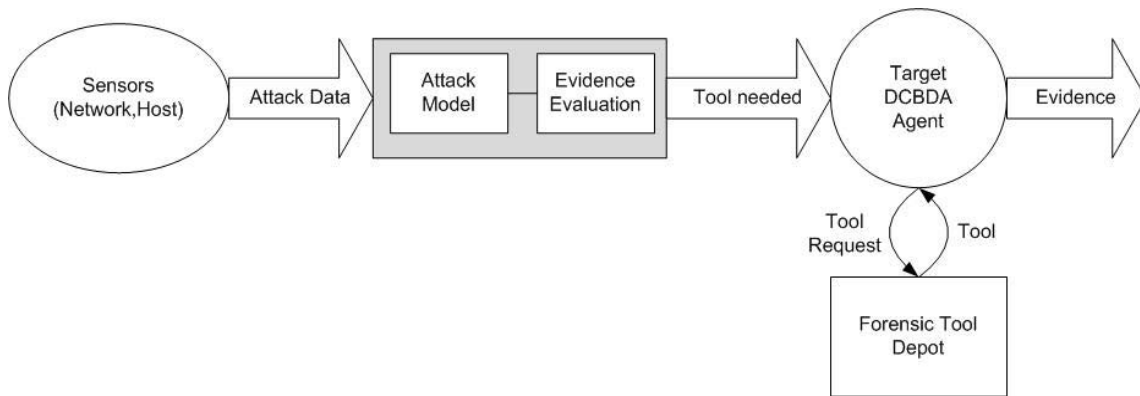
Figure 38. Notional Automated DCBDA Diagram.

Sensor data from a detection system on the asset or the network the asset uses is input into the cyber attack model. This sensor data is then matched in the DCBDA cyber attack tree model to an evidence marker. This marker then identifies a forensic tool the agent can use to retrieve the data from the target. The forensic agent then retrieves the identified tool from a notional Forensic Tool Depot collection of tools and executes the tool, forwarding the results for analysis. This proposed research would need to develop a method to automate the tool selection from the attack data provided from a sensor. This automation would entail attack scenario analysis and evidence evaluation covered in this thesis research.

www.manaraa.com

5.2.4. *Threat Agent Profiling Applied to Attack Tree Analysis*

This future work effort would seek to expand the capabilities based analysis conducted on the attack tree model. Intelligence on a threat agent profile could be applied to the attack tree to further refine selected attack scenarios and evidence markers. This future research area concerns the possible use of studied attack indicators and threat agent profiles to generate pruned attack trees for tailored, efficient and effective forensic analysis [Kle01]. This hacker profiling could also lead to the application of behavior pattern analysis modeling. These behavior profiles may be applied to the attack tree model, as part of the DCBDA process. This can be done by weighing the nodes in the tree with values associated with the profile used. This future work would incorporate threat profile studies, such as the insider threat, to prune the modeled attack tree methodology for a tailored, targeted analysis.

5.2.5. *Defense Tree Model*

Defense tree modeling is closely linked to attack tree modeling. While attack tree modeling focuses on the actions needed to complete an attack root goal, defense modeling centers on the goal of mitigating an attack and the actions taken to support that goal. The focus of this future research area would be to incorporate a defense tree model into the DCBDA process. Related work in this area has shown that a close and beneficial relationship can be shown between the two models [LHS03] [EkS09]. This defense model could be used in conjunction with the attack model to form better evidence markers and as a result work to bolster network defense efforts.

5.3. Concluding Remarks

This research contributed a robust systematic process to capture and utilize the comprehensive knowledge of cyber attack methodology for cyber battle damage assessment. This work is unique from other research in that it focuses on bolstering the capabilities of the technical damage assessment portion of an organization's ability to determine *impact* of malicious cyber events.

115

Appendix A. Attack Vector Categories

Table 38. Attack Vectors Categories [DOD09].

Attack Vector Category Number		Description
1	**Sub-category**	**Reconnaissance**: Information was accessible and used to characterize DOD systems, applications, networks, and users that may be useful in formulating an attack.
	A	Information Gathering and Data Mining: Activity that seeks to gather information from publicly available sources.
	B	Network Scan: Activity that targets multiple IP addresses. This is referred to as a horizontal scan.
	C	System Scan: Activity that targets a single IP address across a range of ports. This is referred to as a vertical scan.
2	**Sub-category**	**Authorized User:** A user with authorized access took specific actions that resulted in jeopardizing DOD systems or data.
	A	Purposeful: An authorized user knowingly took specific actions that jeopardized DOD systems or data.
	B	Accidental: An authorized user took actions that had consequences over and above the intentions and jeopardized DOD systems or data.
3	**Sub-category**	**Social Engineering:** Human interaction (social skills) or deception used to gain access to resources or information.
	A	E-mail: E-mail is the primary vehicle used to deliver a malicious payload or gain access to resources or information.
	B	Web site: A Web site is the primary vehicle used to deliver a malicious payload or gain access to resources or information.
	C	Other: A user was deceived or manipulated in a way that is not covered by the other types of social engineering.

116

Table 39. Attack Vectors Categories Continued [DOD09].

4	**Sub-category**	**Configuration Management:** Compromise resulting from the inadequate or improper configuration of an information system.
	A	Network: A system that provides network-based services was improperly or inadequately configured.
	B	Operating System: An operating system was improperly or inadequately configured.
	C	Application: An application was improperly or inadequately configured.
5	**Sub-category**	**Software Flaw:** A vulnerability in the software that allows for the unauthorized use of or access to an information system in a way that violates the system's security policy.
	A	Exploited New Vulnerability: This vulnerability was unknown prior to the event or there was no mechanism available to prevent it.
	B	Exploited Known Vulnerability: This vulnerability was known prior to the event and there was a mechanism available to prevent it.
6	**Sub-category**	**Transitive Trust:** Compromise resulting from the implicit or explicit trust relationship between security domains.
	A	Other System Compromise: Compromise resulting from access previously gained on another DOD system.
	B	Masquerading: Compromise resulting from the unauthorized use of a valid user's credentials. This may include cryptographic material, account credentials, or other identification information.

Table 40. Attack Vectors Categories Continued [DOD09].

7	**Sub-category**	**Resource Exhaustion:** The consumption of system resources that prevents legitimate users from accessing a resource, service, or information.
	A	Non-Distributed Network Activity: Activity from a single IP address that overwhelms system or network resources. This is generally associated with a DoS incident.
	B	Distributed Network Activity: Activity from multiple IP addresses that overwhelms system or network resources. This is generally associated with a DoS incident.
8	**Sub-category**	**Physical Access:** The unauthorized physical access to resources.
	A	Mishandled or lost resource: Equipment was stolen, lost, or left accessible to unauthorized parties.
	B	Local access to system: An unauthorized user was provided local physical access to a DOD information resource.
	C	Abuse of resources: The physical destruction of an information resource by an unauthorized party.
9	**Sub-category**	Other
	A	New Attack Vector: The attack vector is not covered by the listed methods. Description of the attack vector must be included in the incident comments.
10	**Sub-category**	Unknown
	A	Unable to determine: Attack vector could not be determined with the information available.

Appendix B. CAMEL Attack Action Data

Table 41. CAMEL Attack Action Data.

Attack Action	Parent Action(s)	Child Action(s)	Attack Vector	Attack Results/Goal	Risk of Discover	Impact
Reconnaissance	Cyber Attack	Ping, Whois	9a	This is the initial information gathering action of a cyber attack	L	M
Ping	Reconnaissance		1a	This action attempts to connect to an asset on the target network	L	M
Whois	Reconnaissance		1a	This action enumerations domain information	L	H
Scanning	Cyber Attack	Port Scanning, Network Mapping	1a	This is the active information gathering action for a cyber attack	L	H
Port Scanning	Scanning		1b	This action determines open ports of the assets on a target network	L	H
Network Mapping	Scanning		1b	This action maps the assets of the target network	M	M
Gain Access	Cyber Attack	Password Attack, Web Application Attack	2	This action seeks to gain access to an asset	H	H
Password Attack	Gain Access		2a	This action attempts to use a password for access	L	H
Web Application Attack	Gain Access		3b	This action gains access through a web application	L	H
Maintain Access	Cyber Attack	Backdoor, Trojan	5	This action seeks to maintain access after intial entry is gained	H	H
Backdoor	Maintain Access		5b	This action installs a backdoor for later use by the attacker	H	H
Trojan	Maintain Access		5b	This action leaves embedded attack programs running	L	H
Covering Tracks	Cyber Attack	Log Obfuscation, Attack Data Obfuscation, Persistent Access Obfuscation, Covert Channel	5	This is the action an attacker does to hide unauthorized actions taken on a target system	L	H
Log Obfuscation	Covering Tracks	Fake Log Entries, Erase Logs, Disable Auditing, Overwrite Logs	4	This is the targeted manipulation of a system's logging capability to obfuscate an attack	L	H
Attack Data Obfuscation	Covering Tracks	Hide Attack Data, Delete Attack Files	4	This is the removal of attacker files	L	H

119

Table 42. CAMEL Attack Action Data Continued.

Attack Action	Parent Action(s)	Child Action(s)	Attack Vector	Attack Results/Goal	Risk of Discover	Impact
Persistent Access Obfuscation	Covering Tracks	Rootkit	4	This is the removal of unauthorized persitent access indicators	L	H
Covert Channel	Covering Tracks	ICMP Covert Channel, HTTP Covert Channel, DNS Covert Channel, TCP Covert Channel, 802.11 Covert Tunnel	4a,4b	This action installs a covert channel to obfuscate remote calls to persistent attack files	L	H
Hide Attack Data	Attack File Obfuscation	Steganography, Alternate Data Stream, Time Stamp Alteration, Hide Files/Folders	4b	This action hides attack data on the system from discovery	L	H
Alternate Data Stream	Hide Attack Data		4b	This action forks extended data onto a file, this is used by an attacker to hide attack data	L	H
Steganography	Hide Attack Data		2a	This action hides attack data in non-hidden files	L	H
Time Stamp Alteration	Hide Attack Data		2a	This action modifies the time attribute values for a file	L	H
Hide Files/Folders	Hide Attack Data		2a	This action modifies the hidden flag for a file/folder	H	M
Delete Attack Files	Attack File Obfuscation	Delete Prefetch, Delete Browsing History, Delete Clipboard Data, Delete Shell History, Delete "Recent" History	2a	This action is the deletion of attack tools	L	M
Delete Prefetch	Delete Attack Files		2a	Remove attack data from the prefetch data	L	M
Delete Browsing History	Delete Attack Files		2a	Remove attack data from any browsing history	L	M
Delete Clipboard Data	Delete Attack Files		2a	Remove from the clipboard of any attack data	L	M
Delete Shell History	Delete Attack Files		2a	Remove data from the shell	L	L
Delete "Recent" History	Delete Attack Files		2a	Remove recent activity from the system	M	M
Disable Auditing	Log Obfuscation		2a	Disable system auditing of actions taken	M	H
Overwrite Logs	Log Obfuscation		2a	This action creates events that overwrite events that the attacker wishes to hide/erase	L	H
Fake Log Entries	Log Obfuscation	Delete Log Entry, Add Log Entry	4b	This action creates false log entries to obfuscate attack actions	L	H
Delete Log Entry	Fake Log Entries		4b	This action deletes attack log entries	L	H
Add Log Entry	Fake Log Entries		4b	This action adds to the log entries to hide attack data	L	H
Erase Logs	Log Obfuscation	Erase System Logs, Erase Application Logs, Erase Security Logs	4b	This action erases logs on a target system	H	L

Table 43. CAMEL Attack Action Data Continued.

Attack Action	Parent Action(s)	Child Action(s)	Attack Vector	Attack Results/Goal	Risk of Discover	Impact
Erase System Logs	Erase Logs		4b	This action erases system logs	H	L
Erase Application Logs	Erase Logs		4b	This action erases application logs	H	L
Erase Security Logs	Erase Logs		4b	This action erases security logs	H	L
Rootkit	Persistent Access Obfuscation, Hide Attack Data	Kernel rootkit, User rootkit, Firmware Rootkit, Virtualized Rootkit, Application Rootkit	5a	This action installs a rootkit to hide attack data	L	H
Kernel Rootkit	Rootkit		5a,5b	This action installs a kernel level rootkit to hide attack data	L	H
User Rootkit	Rootkit		5a,5b	This action installs a user level rootkit to hide attack data	L	H
Application Rootkit	Rootkit		5a,5b	This action installs a rootkit in an application	H	M
Virtualized Rootkit	Rootkit		5a,5b	This action installs a rootkit in a virtual hardware	N/A	N/A
Firmware Rootkit	Rootkit		5a,5b	This action installs a rootkit in firmware	L	L
ICMP Covert Channel	Covert Channel		4a,4b	This action installs an ICMP covert channel	M	H
HTTP Covert Channel	Covert Channel		4a,4b	This action installs a HTTP covert channel	H	H
DNS Covert Channel	Covert Channel		4a,4b	This action installs a DNS covert channel	L	H
TCP Covert Tunnel	Covert Channel		4a,4b	This action installs a TCP covert channel	H	L
802.11 Covert Tunnel	Covert Channel		4a,4b	This action installs a wireless covert channel	N/A	N/A

121

Appendix C. CAMEL Attack Method Data

Table 44. CAMEL Attack Method Data.

Attack Method	Attack Action	Attack Method Description	Source
rcovert	802.11 Covert Tunnel	Covert channel using valid ACK frames	http://rfakeap.tuxfamily.org/
cmd: type	Alternate Data Stream	Command line type	http://www.windowsecurity.com/articles/Alternate_Data_Streams.html
cmd: cp	Alternate Data Stream	Command line type	[MSK05]
AFX Windows RootKit	Application Rootkit	System patch to hide information	http://www.megasecurity.org/trojans/a/aphex/Afx_win_rootkit2003.html
Evidence Eliminator	Delete Attack Files	Erase temp files, histories, recent documents	http://www.evidence-eliminator.com/
Tracks Eraser Pro	Delete Attack Files	Erase temp files, histories, recent documents	http://www.acesoft.net/
Netcat	Backdoor	Backdoor remote access program	http://www.securityfocus.com/tools/139
ZeroTracks	Delete browsing history	Erase temp files, histories, recent documents	http://zerotracks.en.softonic.com/
cmd: del windows\prefetch	Delete Prefetch	use shell to delete files	
AuditPol	Disable Auditing	Windows NT Resource Kit for system administrators	http://technet.microsoft.com/en-us/library/cc731451(WS.10).aspx
Disable EventLog service	Disable Auditing	Disable EventLog service on startup	http://support.microsoft.com/kb/172156
Iodine	DNS Covert Tunnel	IPv4 data through a DNS server	http://code.kryo.se/iodine/
DNSCat	DNS Covert Tunnel	bi-directional communication through DNS servers	http://tadek.pietraszek.org/projects/DNScat/
TCP-over-DNS	DNS Covert Tunnel	contains a special dns server and a special dns client	http://analogbit.com/tcp-over-dns_howto
elsave	Erase logs	ELSave is a tool to save and/or clear a NT event log.	http://www.ibt.ku.dk/jesper/ELSave/
ClearLogs	Erase Logs	ClearLogs clears the event log (Security, System or Application) that you specify	http://www.ntsecurity.nu/toolbox/clearlogs/
Evidence Eliminator	Erase Logs	Erase logs	http://www.evidence-eliminator.com/
WinZapper	Fake Log Entries	erase event records selectively	http://www.ntsecurity.nu/toolbox/winzapper/
Alureon	Firmware Rootkit	Win 7 MBR modifications	http://www.virusbtn.com/pdf/conference_slides/2010/Johnson-VB2010.pdf
File hidden option	Hidden option on file	hidden' option is selected for the file	http://www.microsoft.com/windowsxp/using/helpandsupport/learnmore/tips/hiddenfiles.mspx

122

Table 45. CAMEL Attack Method Data Continued.

Attack Method	Attack Action	Attack Method Description	Source
cmd: attrib +h	Hide Files/Folders	cmd: attrib +h	http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/attrib.mspx?mfr=true
Hide Folders 2009	Hide folder	can make your files and folders inaccessible, invisible or protect them from	http://www.fspro.net/hide-folders/
Folder hidden option	Hide folder	hidden' option is selected for the folder	http://windows.microsoft.com/en-US/windows-vista/Show-hidden-files
Hopster	HTTP Covert Tunnel	Client bypasses proxy servers	http://www.hopster.com/
httptunnel	HTTP Covert Tunnel	httptunnel creates a bidirectional virtual data connection tunnelled in HTTP requests	http://www.nocrew.org/software/httptunnel.html
ptunnel	IMCP Covert Tunnel	Tunnel TCP connections to a remote host using ICMP echo request/reply	http://www.neophob.com/2007/10/pingtunnel-for-windows-icmp-tunnel/
Fu	Kernel Rootkit	DKOM to hide processes	https://www.rootkit.com/vault/fuzen_op/FU_README.txt
Cheops-ng	Network Mapping	Network management tool for mapping and monitoring your network	http://cheops-ng.sourceforge.net/
Brutus	Password Attack	Remote password cracker	http://www.hoobie.net/brutus/
ICMP Echo requests	Ping	Send ICMP echo request to target	http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/ping.mspx?mfr=true
nmap	Port Scanning	Port scanning tool	http://nmap.org/
Steghide	Steganography	hide data in various kinds of image- and audio-files	http://steghide.sourceforge.net/
wbStego4open	Steganography	steganography in bitmaps, text files, HTML files and PDF files	http://wbstego.wbailer.com/
Snow	Steganography	conceal messages in ASCII text by appending whitespace	http://www.darkside.com.au/snow/index.html
ncovert	TCP Covert Tunnel	hide network file transfers across the Internet	http://ncovert.sourceforge.net/
Timestomp	Time Stamp Alteration	modifies the time attribute values for a file	http://www.blackhat.com/presentations/bh-usa-05/bh-us-05-foster-liu-update.pdf
Beast	Trojan	Backdoor program	http://h4ck3r.in/board/showthread.php?tid=171

123

Table 46. CAMEL Attack Method Data Continued.

Attack Method	Attack Action	Attack Method Description	Source
Vanquish	User Rootkit	Processes, Handles, Modules, Files & Folders, Registry Values, Services	https://www.rootkit.com/vault/xshadow/ReadMe.txt
Hacker Defender	User Rootkit	User rootkit to hide processes	http://www.rootkit.com/board_project_fused.php?did=proj5
Nuclear	User Rootkit	Hide processes, directories, registry, connections	http://www.nuclearwintercrew.com/Products-View/63/Nuclear_Rootkit_1.0/
Blue Pill	Virtualized Rootkit	x86 virtualization rootkit	http://theinvisiblethings.blogspot.com/2006/06/introducing-blue-pill.html
WebScarab	Web Application Attack	Analyze HTTP applications	http://www.owasp.org/index.php/Category:OWASP_WebScarab_Project
Samspade	Whois	Network query tool	http://www.pcworld.com/downloads/file/fid,4709-order,1-page,1-c,alldownloads/description.html

124

Appendix D. CAMEL Evidence Analysis Data

Table 47. CAMEL Evidence Analysis Data.

Attack Action / Method	Forensic Marker	Forensic Tools	Evidence Confidence	Evidence Source
ICMP Echo requests	ICMP ping incoming and outgoing	Wireshark	H	http://www.wireshark.org/
Samspade	Ping, SMTP VFRY, web site activity	Wireshark	M	http://majorgeeks.com/Sam_Spade_d594.html
nmap	Port scanning, OS enum	Wireshark	H	http://nmap.org/
Cheops-ng	Port scanning, OS enum	Wireshark	M	http://cheops-ng.sourceforge.net/screenshots.php
Brutus	Failed login attempts in log	PyFlag	H	http://www.webhostgear.com/240.html
WebScarab	Malformed data	Wireshark	H	http://www.owasp.org/index.php/Category:OWASP_WebScarab_Project
Netcat	suspicious open ports	netstat	H	http://technet.microsoft.com/en-us/library/bb490947.aspx
Beast	Suspicious Trojan activity	Regedit.exe	H	http://www.exterminate-it.com/malpedia/remove-beast#howfiles
cmd: del windows\prefetch	windows\prefetch missing	cmd	H	
Hide Folders 2009	Hidden folders	cmd	L	http://www.fspro.net/hide-folders/
Folder hidden option	Hidden folders	cmd	L	http://www.microsoft.com/windowsxp/using/helpandsupport/learnmore/tips/hiddenfiles.mspx
File hidden option	Hidden files	cmd	L	http://www.microsoft.com/windowsxp/using/helpandsupport/learnmore/tips/hiddenfiles.mspx
Steghide	Suspicious files/sizes	Stegdetect	H	http://www.outguess.org/detection.php

125

Table 48. CAMEL Evidence Analysis Data Continued.

Attack Action / Method	Forensic Marker	Forensic Tools	Evidence Confidence	Evidence Source
wbStego4open	Suspicious files/sizes	Hashkeeper	M	http://www.garykessler.net/library/fsc_stego.html
Snow	Suspicious files/sizes	Stegdetect	H	http://www.outguess.org/detection.php
cmd: type	Suspicious files/sizes	Streams, ADSTools, ADS Scanner, ADS Spy	L	http://technet.microsoft.com/en-us/sysinternals/bb897440.aspx
cmd: cp	Suspicious files/sizes	Streams, ADSTools, ADS Scanner, ADS Spy	L	http://technet.microsoft.com/en-us/sysinternals/bb897440.aspx
cmd: attrib +h	Hidden files	cmd	H	http://www.microsoft.com/windowsxp/using/helpandsupport/learnmore/tips/hiddenfiles.mspx
Timestomp	Suspicious time MACE values	dir c:\ /A /S /T:W	M	http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/ntcmds.mspx?mfr=true
ZeroTracks	Files removed	Recuva, File Scavenger	M	http://www.kleinsoft.co.za/products.html
Evidence Eliminator	Files removed	Recuva, File Scavenger	M	http://windows.microsoft.com/en-US/windows7/Recover-lost-or-deleted-files
Tracks Eraser Pro	Files removed	Recuva, File Scavenger	M	http://windows.microsoft.com/en-US/windows7/Recover-lost-or-deleted-files
AuditPol	Missing/incomplete logs	Event Viewer	H	http://technet.microsoft.com/en-us/library/cc766042.aspx
Disable EventLog service	Disabled EventLog service	Services	H	http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/cpanel_admintools.mspx?mfr=true

126

Table 49. CAMEL Evidence Analysis Data Continued.

Attack Action / Method	Forensic Marker	Forensic Tools	Evidence Confidence	Evidence Source
Evidence Eliminator	Missing/incomplete logs	Event Viewer	H	http://windows.microsoft.com/en-US/windows7/Recover-lost-or-deleted-files
WinZapper	Missing/incomplete logs	Event Viewer	H	http://msdn.microsoft.com/en-us/library/aa385780(v=vs.85).aspx
ClearLogs	Missing/incomplete logs	Event Viewer	H	http://msdn.microsoft.com/en-us/library/aa385780(v=vs.85).aspx
Vanquish	VANQUISH.DLL	RootkitRevealer	H	http://www.rootkit.com/newsread.php?newsid=35
Hacker Defender	hxdef.exe	RootkitRevealer	H	http://www.carnal0wnage.com/papers/rootkit_for_the_masses.pdf
Nuclear	c:\WINDOWS\nkit.dll	RootkitRevealer	H	http://www.nuclearwintercrew.com/Products-Screenshot/63/Nuclear_Rootkit_1.0/
Fu	PspCidTable	RootkitRevealer, F-Secure Blacklight	H	[DBM09]
DNS2TCP	suspicious traffic over port 53	Wireshark	H	http://www.hsc.fr/ressources/outils/dns2tcp/index.html.en
Iodine	suspicious traffic over port 53	Wireshark	H	http://www.securitywire.com/
Iodine	TAP32 driver	cmd	H	
DNSCat	suspicious DNS, Java 1.4+ installed	Wireshark	H	http://tadek.pietraszek.org/projects/DNScat/
TCP-over-DNS	suspicious traffic over port 53	Wireshark	H	http://analogbit.com/tcp-over-dns_howto
ICMP Covert Channel	suspicious ICMP traffic	Wireshark	H	http://www.cs.uit.no/~daniels/PingTunnel/

127

Table 50. CAMEL Evidence Analysis Data Continued.

Attack Action / Method	Forensic Marker	Forensic Tools	Evidence Confidence	Evidence Source
ptunnel	ptunnel.exe on system	cmd/search	M	
ptunnel	WinPCap installed	cmd/search	H	http://www.neophob.com/2007/10/pingtunnel-for-windows-icmp-tunnel/
ncovert	SIN as data field	Wireshark	H	http://www.blackhat.com/presentations/bh-usa-03/bh-us-03-simplenomad/bh-us-03-simplenomad.pdf
Hopster	Suspicious HTTP traffic	Wireshark	H	http://www.hopster.com/
httptunnel	Suspicious HTTP traffic	Wireshark	H	http://www.nocrew.org/software/httptunnel.html
rcovert	Suspicious activity	Wireshark	H	http://rfakeap.tuxfamily.org/#Raw_Covert
Hans	Suspicious ICMP traffic	Wireshark	H	http://code.gerade.org/hans/
Skeeve	Suspicious ICMP traffic	Wireshark	H	http://www.gray-world.net/poc_skeeve.shtml
ICPMTX	Suspicious ICMP traffic	Wireshark	H	http://thomer.com/icmptx/
Alureon	Suspicious activity	RootkitRevealer	H	http://technet.microsoft.com/en-us/sysinternals/bb897445
Blue Pill	Suspicious activity	cmd	L	http://theinvisiblethings.blogspot.com/2006/06/introducing-blue-pill.html
AFX Windows RootKit	Suspicious activity	RootkitRevealer	H	http://technet.microsoft.com/en-us/sysinternals/bb897445
DNS Covert Tunnel	Suspicious traffic over port 53	Wireshark	H	http://www.wireshark.org/
Rootkit	Suspected Rootkit Activity	RootkitRevealer, F-Secure Blacklight	H	
Alternate Data Stream	System search for all active ADS	Streams, ADSTools, ADS Scanner, ADS Spy	H	http://www.pointstone.com/products/ADS-Scanner/

Appendix E. CAMEL Tool and Method Data

Table 51. CAMEL Tool and Method Data.

Forensic Marker	Forensic Tool/Method	Tool Confidence	Tool/Method Source
ICMP ping incoming and outgoing	Wireshark	H	http://www.wireshark.org/
Ping, SMTP VFRY, web site activity	Wireshark	H	http://www.wireshark.org/
Port scanning, OS enum	Wireshark	H	http://www.wireshark.org/
Port scanning, OS enum	Wireshark	H	http://www.wireshark.org/
Failed login attempts in log	PyFlag	H	http://www.pyflag.net/cgi-bin/moin.cgi
Malformed data	Wireshark	H	http://www.wireshark.org/
suspicious open ports	netstat	M	http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/netstat.mspx?mfr=true
Suspicious Trojan activity	Regedit.exe	H	http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/tools_regeditors.mspx
\windows\system32\config missing	cmd	H	
windows\prefetch missing	cmd	H	
Suspicious files/sizes	Stegdetect	H	http://www.outguess.org/detection.php
Hidden folders	cmd	H	
Suspicious files/sizes	StegSpy	H	http://www.spy-hunter.com/stegspydownload.htm
Suspicious files/sizes	Hashkeeper	M	http://www.justice.gov/ndic/domex/hashkeeper.htm
Hidden files	cmd	H	
Suspicious time Modified, Access, and Creation values	dir c:\ /A /S /T:W	H	
Files removed	Recuva	M	http://www.piriform.com/recuva

129

Table 52. CAMEL Tool and Method Data Continued.

Forensic Marker	Forensic Tool/Method	Tool Confidence	Tool/Method Source
VANQUISH.DLL	RootkitRevealer	H	http://technet.microsoft.com/en-us/sysinternals/bb897445
hxdef.exe	RootkitRevealer	H	http://technet.microsoft.com/en-us/sysinternals/bb897445
c:\WINDOWS\nkit.dll, c:\WINDOWS\Rootkit.exe	RootkitRevealer	H	http://technet.microsoft.com/en-us/sysinternals/bb897445
PspCidTable	RootkitRevealer	H	http://technet.microsoft.com/en-us/sysinternals/bb897445
suspicious traffic over port 53	Wireshark	H	http://www.wireshark.org/
suspicious ICMP traffic	Wireshark	H	http://www.wireshark.org/
ptunnel.exe on system	cmd/search	M	
WinPCap installed	cmd/search	H	
SIN as data field	Wireshark	H	http://www.wireshark.org/
Suspicious HTTP traffic	Wireshark	H	http://www.wireshark.org/
Suspicious activity	Wireshark	H	http://www.wireshark.org/
Suspicious activity	RootkitRevealer	H	http://technet.microsoft.com/en-us/sysinternals/bb897445
Suspicious traffic over port 53	Wireshark	H	http://www.wireshark.org/
Suspicious traffic over port 53	Wireshark	H	http://www.wireshark.org/
Suspicious activity	RootkitRevealer	H	http://technet.microsoft.com/en-us/sysinternals/bb897445
Streams will examine the files and directories for streams	Streams	M	http://technet.microsoft.com/en-us/sysinternals/bb897440.aspx
Searches drives & lists all files that have ADS	ADSTools	M	http://www.soft32.com/download_207535.html
System search for all active ADS	ADS Scanner	M	http://www.pointstone.com/products/ADS-Scanner/
List, view or delete Alternate Data Streams (ADS)	ADS Spy	M	http://www.brothersoft.com/ads-spy-74079.html

130

www.manaraa.com

Table 53. CAMEL Tool and Method Data Continued.

Forensic Marker	Forensic Tool/Method	Tool Confidence	Tool/Method Source
Suspected Rootkit activity	RootkitRevealer	H	http://technet.microsoft.com/en-us/sysinternals/bb897445
Suspected Rootkit activity	F-Secure Blacklight	H	
deleted files			http://www.quetek.com/prod02.htm
deleted files	Recycle Bin		

Appendix F. CAMEL Attack Action Metrics

Table 54. CAMEL Attack Action Metrics.

Attack Action	Risk of Discovery	Impact
Reconnaissance	L	M
Ping	L	M
Whois	L	H
Scanning	L	H
Port Scanning	L	H
Network Mapping	M	M
Gain Access	H	H
Password Attack	L	H
Web Application Attack	L	H
Maintain Access	H	H
Backdoor	H	H
Trojan	L	H
Covering Tracks	L	H
Log Obfuscation	L	H
Attack Data Obfuscation	L	H
Persistent Access Obfuscation	L	H
Covert Channel	L	H
Hide Attack Data	L	H
Alternate Data Stream	L	H
Steganography	L	H
Time Stamp Alteration	L	H
Hide Files/Folders	H	M
Delete Attack Files	L	M
Delete Prefetch	L	M
Delete Browsing History	L	M
Delete Clipboard Data	L	M

Table 55. CAMEL Attack Action Metrics Continued.

Attack Action	Risk of Discovery	Impact
Delete Shell History	L	L
Delete "Recent" History	M	M
Disable Auditing	M	H
Overwrite Logs	L	H
Fake Log Entries	L	H
Delete Log Entry	L	H
Add Log Entry	L	H
Erase Logs	H	L
Erase System Logs	H	L
Erase Application Logs	H	L
Erase Security Logs	H	L
Rootkit	L	H
Kernel Rootkit	L	H
User Rootkit	L	H
Application Rootkit	H	M
Virtualized Rootkit	N/A	N/A
Firmware Rootkit	L	L
ICMP Covert Channel	M	H
HTTP Covert Channel	H	H
DNS Covert Channel	L	H
TCP Covert Tunnel	H	L
802.11 Covert Tunnel	N/A	N/A

133

Appendix G. CAMEL Creation Outline

CAMEL Creation Outline

Create and model the Cyber Attack Methodology Exhaustive List (CAMEL). CAMEL is comprised of attack actions, methods, metrics, evidence markers and the associated forensic tools.

1. <u>Gather attack action data</u>

    This is the attacker methodology detailed as a list of actions taken to achieve a goal action.

    a. Name of action

    b. Parent action

    c. Child action

    d. Attack results

    e. Other (Attack vector, etc.)

2. <u>For each action collected, list all attack methods (tools, TTPs)</u>

    These are the methods used to create the action. This is the procedure to produce the act.

    a. Attack method

    b. Attack Action for method

    c. Attack category (Attack action method used for)

    d. Attack method description

    e. Attack method source

3. <u>For each action and method, list all evidence which identify it</u>

    This is the detailed fingerprinting of what the act and/or method 'does' when it transpires on the target system.

    a. Name of evidence

    b. Attack action/method the evidence identifies

     c. Forensic markers which will be collected

     d. Source for evidence marker

4. <u>For each evidence marker, list the associated tool to harvest the marker</u>

This step details the forensic tools/TTPs which can collect the associated marker.

     a. Name of forensic marker

     b. Name of evidence it identifies

     c. Forensic tool to collect marker

     d. Source for forensic tool

5. <u>Assign metrics to each attack action</u>

Each action now has attack method, evidence and collection method data associated with it. The capabilities of an action, as determined by the organization using CAMEL, can now be assigned as metrics.

     a. Determine metrics for action

     b. Assign values to metrics

6. <u>Model the action, method, marker and metric in an attack tree</u>

This is the graphical representation of the data collected in steps 1-5. This model is referred to as the Cyber Attack Methodology Attack Tree (CAMAT).

135

<center>**Appendix H. Test Systems Setup**</center>

Test_1

1. WinPCap 4.1.2 installed.

The program WinPcap is downloaded from http://www.winpcap.org/. Run the installer file WinPcap_4_1_2.exe on the system (next, next, I agree, install).

2. ptunnel.exe used to create an ICMP tunnel to a receiving proxy previously setup.

Download *ptun-rel1.zip* from http://www.neophob.com/2007/10/pingtunnel-for-windows-icmp-tunnel/ on both target and a designated proxy server. Running *ptunnel.exe -h* will list help to include examples.

Server:

A separate system is designated as a server. This will act as the proxy, listening for and handling the incoming ptunnel icmp packets. The server must run the following command with the device changed to the specific device of the system.

ptunnel -v 4 -c "\Device\NPF_{EED408B.....}"

Client:

The victim system is the where the covert channel is installed and used to bypass firewalls. The following command is run with options changed as appropriate:

ptunnel -p <SERVERIP> -lp 8000 -da <WEB-PROXY> -dp <PORT> -v 4

<center>136</center>

ServerIP is the IP of the server listening for the connections.

Web-proxy is the web proxy of the local network.

Port is the port to be used on the client (80 for web)

Now a web browser can be set up to use the running ptunnel port as the proxy server (127.0.0.1 for IP and the Port setup on the ptunnel command) connection for internet access. SSH can also setup and used.

3. **The FU rootkit used to hide the ptunnel.exe process.**

Download the FU_rootkit from www.rootkit.com. Find the process ID (PID) of the ptunnel.exe using Windows Task Manager. Run the following command on the fu.exe file in the FU_rootkit EXE folder. This will hide the putnnel.exe process.

fu -ph (PID)

4. **Evidence Eliminator used to clear the event logs of any trace of these actions.**

Download the Evidence Eliminator version 6.03 tool from http://www.evidence-eliminator.com/downloads.d2w. The install file downloaded will be called *insteelmd.exe*. Run the install program agreeing (next, YES, next, finish). This will install Evidence Eliminator on the system. Now run the program. This experiment assumes the retail version of the software is purchased. This program will clear evidence of Recycle Bin, Application Logs, Temp Files, Internet Explorer cache, and Clipboard data.

137

Test_2

1. The EventLogs service disabled in the Administrator tool: Services.

The Administrator tool Services is used to change the Startup type of Event Log

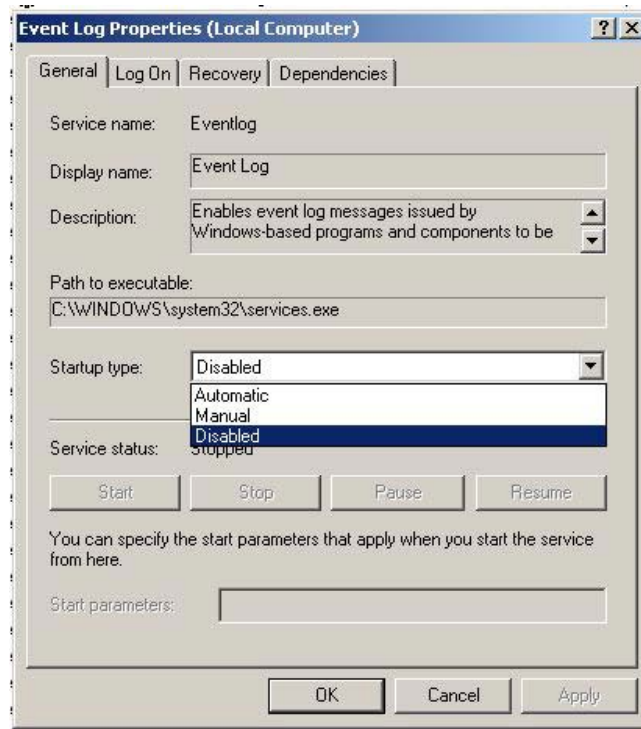to *disabled*. Figure 39 shows the option that is changed to disable the service.

Figure 39. Test_2 Setup Event Log Service.

2. Two Alternate Data Streams (ADS) created using the type cmd to hide notional attack executable bad.exe in calc.exe.

A malicious program (in this case a renamed copied executable file from the

system) *bad.exe* is put in the alternate data stream of *calc.exe*. Then *calc.exe* is hidden in

notepad.exe thus creating two alternate data streams. The following commands are used.

138

type bad.exe > calc.exe:bad.exe

type calc.exe > notepad.exe:calc.exe

3. Two txt files created, deleted and 'emptied' from Recycle Bin.

Two text files, *Company Sales Plan,txt* and *Company Secrets.txt* are created with notional data in the files. These files are then deleted. The Recycle Bin then ran the "Empty Recycle Bin" option.

4. ZeroTracks used to remove browsing and recent file history.

The program *SoftonicDownloader_for_zerotracks.exe* is downloaded from the site: http://zerotracks.en.softonic.com. The installer is executed, accepting and agreeing to all install options. The installer then downloads the software for installation. The Setup wizard then runs. The wizard is completed using all default selections (next and install).

Once installed, the ZeroTracks program is run. The options in ZeroTracks *Windows Recent Docs* and *Internet Explorer Cache and History* are used to clear (remove selected items) suspect associated files. Figures 40 and 41 show the ZeroTracks window used to delete the browsing and recent file history.
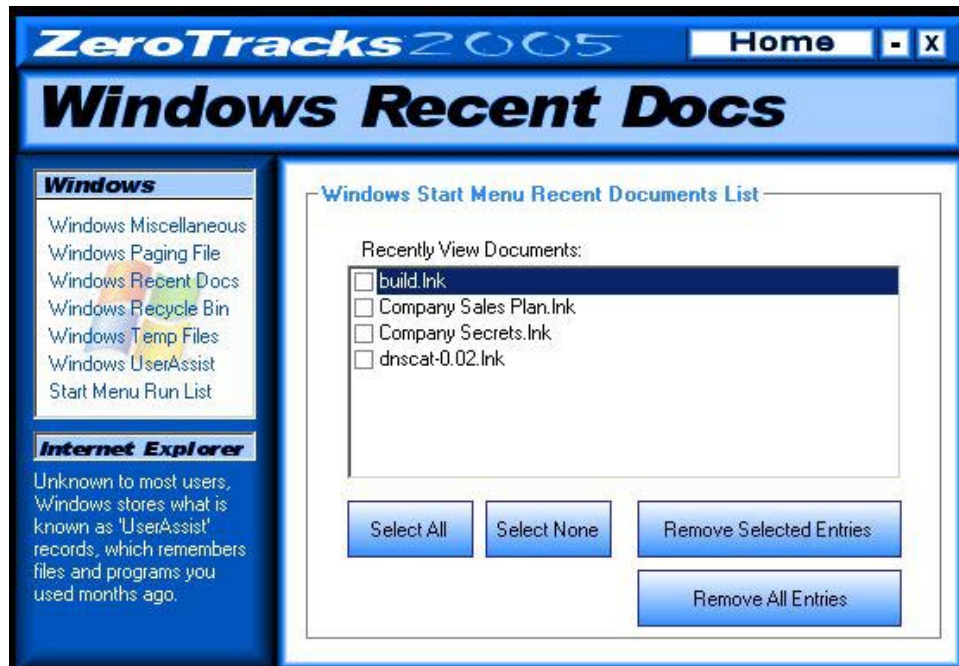
139

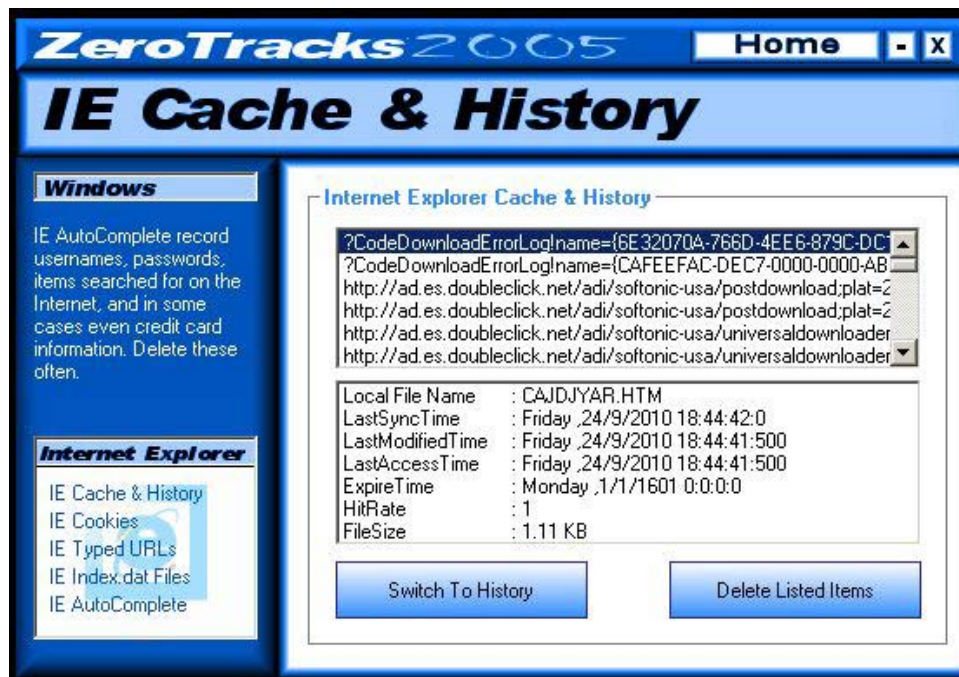Figure 40. Test_2 Setup Zero Tracks Clear Windows Recent Docs.

Figure 41. Test_2 Setup Zero Tracks Clear IE Cache & History.

140

www.manaraa.com

Test_3 Setup

1. **Create secrets.txt file**

For this step of the test setup, create a text file to hold data to be exfiltrated. In this case, secrets.txt was created with notional data in the file.

2. **Use steghide-0.5.1-win32 to hide secrets.txt into winter.jpg.**

Use the program steghide-0.5.1-win32, retrieved from http://steghide.sourceforge.net to hide the secrets.txt file in a jpg file. In this case the file Winter.jpg file was used. The following steghide command was used.

steghide embed -cf Winter.jpg -ef secrets.txt

A passphrase was given and secrets.txt was embedded into Winter.jpg via the steganography tool steghide.

3. **Use timestomp.exe to modify time value of winter.jpg to Monday 1/1/2001 01:01:01 AM**

Timestomp was downloaded from http://www.metasploit.com/data/antiforensics/timestomp.exe. The following command was executed using the timestomp tool to modify Winter.jpg file time attributes.

timestomp Winter.jpg -z "Monday 1/01/2001 01:01:01 AM"

141

Bibliography

[Ame04]   Amenaza, (2004, May), "Understanding Risk Through Attack Tree Analysis,"
            Amenaza Technologies Limited, [Online], Accessed 9 Dec 2010,
            http://www.amenaza.com/downloads/docs/Methodology.pdf.

[Ame05]   Amenaza, (2005, Nov), "Fundamentals of Capabilities-based Attack Tree
            Analysis," Amenaza Technologies Limited, [Online], Accessed 9 Dec 2010,
            http://www.amenaza.com/downloads/docs/AttackTreeFundamentals.pdf.

[Amo94]   E. G. Amoroso, (1994), *Fundamentals of Computer Security Technology*,
            Prentice-Hall Inc., Upper Saddle River, NJ, USA.

[Bos02]   S. Bosworth, (2002), *Computer Security Handbook* (4th ed.), M. E. Kabay
            (Ed.), John Wiley & Sons, Inc., New York, NY, USA.

[Coh87]   F. Cohen, (1987, Feb), "Computer Viruses Theory and Experiments,"
            Computers and Security, Volume 6, Issue 1, pp. 22-35.

[DAF08]   Department of the Air Force, (2008, Mar), Predictive Battlespace Awareness:
            Air and Space Intelligence Preparation of the Operational Environment, AFP 14-
            118, Washington: HQ USAF,
            http://www.e-publishing.af.mil/shared/media/epubs/AFPAM14-118.pdf.

[DBM09]   M. Davis, S. Bodmer and A. LeMasters, (2009, Nov), *Hacking Exposed:
            Malware & Rootkits Secrets & Solutions*, McGraw-Hill, Inc., New York, NY,
            USA.

[DOD07]   Department of Defense, (2007, Jun), *Joint Intelligence,* Joint Publication 2-0,
            http://www.fas.org/irp/doddir/dod/jp2_0.pdf.

[DOD09]   Department of Defense, (2009, Jun), *Information Assurance and Computer
            Network Defense Volume I (Incident Handling Program),* Chairman of the Joint
            Chiefs of Staff Manual 6510.01A,
            http://www.dtic.mil/cjcs_directives/cdata/unlimit/m651001_v1.pdf .

[DOJ04]   U.S. Department of Justice, (2004, Apr), *Forensic Examination of Digital
            Evidence: A guide for Law Enforcement*, Office of Justice Programs, National
            Institute of Justice, Washington, DC.,
            http://www.ncjrs.gov/pdffiles1/nij/199408.pdf .

[Dru10]   D. Drummond, (2010, Dec), "A new approach to China," The Official Google
            Blog, [Online], Accessed 14 Dec 2010,
            http://googleblog.blogspot.com/2010/01/new-approach-to-china.html.

[Edg07]   K. Edge, (2007, Jul), *A Framework For Analyzing And Mitigating The Vulnerabilities Of Complex Systems Via Attack And Protection Trees,* Air Force Institute of Technology (AU), Wright Patterson AFB, OH, http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA472310.

[EkS09]   M. Ekstedt and T. Sommestad, (2009, Mar), "Enterprise architecture models for cyber security analysis," *Power Systems Conference and Exposition*, 2009, PSCE '09, IEEE/PES , pp.1-6, 15-18, doi: 10.1109/PSCE.2009.4840267, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4840267&isnumber=4839920.

[FMC10]   N. Falliere, L. Murchu, and E. Chien, (2010, Nov), "W32.Stuxnet Dossier," Symantec Security Response, [Online], Accessed 14 Dec 2010, http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf.

[For07]   L. Fortson, (2007, Mar), "Towards The Development Of A Defensive Cyber Damage And Mission Impact Methodology," MS thesis, AFIT/GIR/ENV/07-M9, School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA467549&Location=U2&doc=GetTRDoc.pdf.

[GiM02]   J. Giordano and C. Maciag, (2002), "Cyber Forensics: A Military Operations Perspective," *International Journal of Digital Evidence*, Volume 1, Issue 2, http://www.utica.edu/academic/institutes/ecii/publications/articles/A04843F3-99E5-632B-FF420389C0633B1B.pdf.

[HoL98]   J. D. Howard, T. A. Longstaff, (1998, Oct), "A Common Language for Computer Security Incidents," Technical report, Sandia National Laboratories, www.cert.org/research/taxonomy_988667.pdf.

[Hor99]   M. Horony, (1999, Dec), *Information System Incidents: The Development Of A Damage Assessment Model*, MS thesis, AFIT/DIR/LAS/99D-5, School of Engineering and Management, Air Force Institute of Technology (AU), Wright Patterson AFB, OH, http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA374167&Location=U2&doc=GetTRDoc.pdf.

[How97]   J. D. Howard, (1997, Apr), *An Analysis Of Security Incidents On The Internet,* PhD Dissertation, Carnegie Mellon University, Pittsburg, PA, www.cert.org/archive/pdf/JHThesis.pdf.

143

[KaP07]   D. R. Karrels and G. L. Peterson, (2007, Nov), "CyberCraft: Protecting Air
            Force Electronic Systems with Lightweight Agents," Vir V. Phoha and S.S.
            Iyengar (editors), Proceedings of the Cyberspace Research Workshop, pp. 58-62,
            United States Air Force, Shreveport, LA.

[Kar05]   Karppinen, K, (2005), "Security Measurement Based on Attack Trees in a
            Mobile Ad Hoc Network Environment," [Online], Accessed 7 Dec 2010,
            http://www.vtt.fi/inf/pdf/publications/2005/P580.pdf.

[Kle01]   L, Kleen, (2001, Mar), *Malicious Hackers: A Framework For Analysis*, MS
            thesis, AFIT/GOR/ENS/10M-09, School of Engineering and Management, Air
            Force Institute of Technology (AU), Wright Patterson AFB , OH,
            http://www.dtic.mil/cgi-
            bin/GetTRDoc?AD=ADA392952&Location=U2&doc=GetTRDoc.pdf.

[LBM93]   C. Landwehr, A. Bull, J. McDermott, and W. Choi, (1993, Nov), *A Taxonomy of
            Computer Program Security Flaws, with Examples*, Information Technology
            Division, Naval Research Laboratory, Washington, D. C.,
            http://www.dtic.mil/cgi-
            bin/GetTRDoc?AD=ADA274500&Location=U2&doc=GetTRDoc.pdf.

[LHS03]   S. Lathrop, J. M. D. Hill, and J. R. Surdu, (2003, May), "Modeling Network
            Attacks," 12th Conference On Behavior Representation In Modeling And
            Simulation (BRIMS 2003), Scottsdale, Arizona,
            http://www.bucksurdu.com/Professional/Documents/BRIMSMAADNET.pdf.

[Mca10]   McAfee, (2010, Dec), Operation Aurora, [Online], Accessed 7 Dec 2010,
            http://www.mcafee.com/us/threat_center/operation_aurora.html.

[MRH10]   A, Matrosov, E. Rodionov, D. Harley, and J. Malcho, (2010, Dec), "Stuxnet
            Under the Microscope," Revision 1.31, [Online], Accessed 7 Dec 2010,
            http://www.eset.com/resources/white-papers/Stuxnet_Under_the_Microscope.pdf.

[MSK05]   S. McClure, J. Scambray, and G. Kurtz, (2005), *Hacking Exposed: Network
            Security Secrets & Solutions,* 5th ed., McGraw-Hill/Osborne, Inc., New York,
            NY, USA.

[NIS06]   National Institute of Standards and Technology, (2006, Aug), *Guide to
            Integrating Forensic Techniques into Incident Response*, NIST Special
            Publication 800-86, Gaithersburg, MD,
            http://csrc.nist.gov/publications/nistpubs/800-86/SP800-86.pdf.

144

[NIS08] National Institute of Standards and Technology, (2008, Jul), *Performance Measurement Guide for Information Security,* NIST Special Publication 800-85 Revision 1, Gaithersburg, MD, http://csrc.nist.gov/publications/nistpubs/800-55-Rev1/SP800-55-rev1.pdf.

[PoR07] N. Poolsappasit and I. Ray, (2007), "Investigating Computer Attacks using Attack Trees," Advances in Digital Forensics III, I FIP, Volume 242, pp. 331-343, Springer Boston.

[Sch99] B. Schneier, (1999, Dec), "Modeling Security Threats," Dr. Dobbs Journal, [Online], Accessed 9 Dec 2010, http://www.schneier.com/paper-attacktrees-ddj-ft.html.

[SkL05] E. Skoudis and T. Liston, (2005), *Counter Hack Reloaded: A Step-By-Step Guide to Computer Attacks and Effective Defenses*, 2nd ed., Prentice Hall PTR, Upper Saddle River, NJ, USA.

[Thi05] L, Thiem, (2005, Mar), *A study to determine damage assessment methods or models on Air Force networks,* MS thesis, AFIT/GIR/ENV/05M-18, School of Engineering and Management, Air Force Institute of Technology (AU), Wright Patterson AFB, OH, http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA435179&Location=U2&doc=GetTRDoc.pdf.

[UnP02] J. Undercoffer and J. Pinkston, (2002, Oct), *Modeling Computer Attacks: A Target-Centric Ontology for Intrusion detection*, Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, http://www.cs.umbc.edu/csee/research/cadip/2002Symposium/Ont-for-IDS.pdf.

REPORT DOCUMENTATION PAGE

1. REPORT DATE *(DD-MM-YYYY)* 25-03-2011	2. REPORT TYPE Master's Thesis	3. DATES COVERED *(From – To)* Jun 2009 - Mar 2011

4. TITLE AND SUBTITLE Defensive Cyber Battle Damage Assessment Through Attack Methodology Modeling	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Ryan T. Ostler, Capt, USAF	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB, OH 45433-7765	8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GCO/ENG/11-11
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Due to the growing sophisticated capabilities of advanced persistent cyber threats, it is necessary to understand and accurately assess cyber attack damage to digital assets. This thesis proposes a Defensive Cyber Battle Damage Assessment (DCBDA) process which utilizes the comprehensive understanding of all possible cyber attack methodologies captured in a Cyber Attack Methodology Exhaustive List (CAMEL). This research proposes CAMEL to provide detailed knowledge of cyber attack actions, methods, capabilities, forensic evidence and evidence collection methods. This product is modeled as an attack tree called the Cyber Attack Methodology Attack Tree (CAMAT). The proposed DCBDA process uses CAMAT to analyze potential attack scenarios used by an attacker. These scenarios are utilized to identify the associated digital forensic methods in CAMEL to correctly collect and analyze the damage from a cyber attack. The results from the experimentation of the proposed DCBDA process show the process can be successfully applied to cyber attack scenarios to correctly assess the extent, method and damage caused by a cyber attack

15. SUBJECT TERMS
Battle Damage Assessment, Cyber Attack, Digital Forensics

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Dr. Barry E. Mullins AFIT/ENG
REPORT U	ABSTRACT U	c. THIS PAGE U	UU	153	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636 x7979 barry.mullins@afit.edu